CAPACITATED LOT-SIZE SEQUENCING PROBLEM

ZUHAIMY HJ. ISMAIL

RESEARCH VOTE NO:
75094

JABATAN MATEMATIK
FAKULTI SAINS
UNIVERSITI TEKNOLOGI MALAYSIA

2007

Lampiran 20

UTM/RMC/F/0024 (1998)

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN
LAPORAN AKHIR PENYELIDIKAN

TAJUK PROJEK : **CAPACITATED LOT-SIZE SEQUENCING PROBLEM**

Saya **ZUHAIMY HJ. ISMAIL**
(HURUF BESAR)

Mengaku membenarkan **Laporan Akhir Penyelidikan** ini disimpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut :

1. Laporan Akhir Penyelidikan ini adalah hakmilik Universiti Teknologi Malaysia.
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan rujukan sahaja.
3. Perpustakaan dibenarkan membuat penjualan salinan Laporan Akhir Penyelidikan ini bagi kategori TIDAK TERHAD.
4. * Sila tandakan (/)

☐	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau Kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972).
☐	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh Organisasi/badan di mana penyelidikan dijalankan).
☐	TIDAK TERHAD	

TANDATANGAN KETUA PENYELIDIK

DR. ZUHAIMY HJ. ISMAIL
Professor Madya
Jabatan Matematik, Fakulti Sains
Universiti Teknologi Malaysia
Skudai, Johor

Nama & Cop Ketua Penyelidik

Tarikh : 11 Jun 2007

CATATAN : **Jika Laporan Akhir Penyelidikan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan sebagai SULIT dan TERHAD.*

Preface

This report entitle "Modelling of Basic Sequencing Problem in Travelling Salesman Problem and Capacitated Lot-Size Sequencing Problem" has been prepared by Associate Professor Dr. Zuhaimy Hj. Ismail during the project period January 2004 to Jun 2006 at the Department of Mathematics, Universiti Teknologi Malaysia, Skudai Johor.

This report is submitted as the requirement for the completion of Fundamental Research Grant Project. The work was fully supported by The Research Management Centre (RMC), Universiti Teknologi Malaysia. The subject of the report is the study of some methodology in basic scheduling problems. The objective of this study is to study the fundamental concept of Genetic Algorithm (GA) as a mathematical tool in the search in the optimization of scheduling problems. It also explores a few models on a multi-level capacitated lot-size sequencing problem with bottlenecks. I would like to express my gratitude to RMC for their belief and interest in this project.

Next, I would like to thank the Department of Mathematics, UTM for all the support and services provided in making this research a success. This department has provided the necessary platform for the development of the algorithms in this study.

I would like to thank my students and colleague Wan Rohaizad Wan Ibrahim, Szu Za Nah, Khairil Asmani Mahpol and Irhamah for their contribution and devotion in this project. Also, the staff – the academic as well as the administrative at the Department and Faculty of Science, Universiti Teknologi Malaysia

Assoc. Prof. Dr. Zuhaimy Hj. Ismail
Department of Mathematics

Summary

The constraint satisfaction problems are in fact constraint satisfaction sequencing problems, where the aim is to find a sequence for a domain of values such that all the constraints on the sequence are satisfied. We study different aspect of sequencing problem namely the basic sequencing problem in Job Shop Sequencing Problem which is a common sequencing problem in practice. This study attempt to formulate the problem using a sequencing formulation and solving this problem using search techniques developed for the generic constraint satisfaction problem.

The advantages of developing constraint satisfaction search algorithms specifically for sequencing problems are twofold. Firstly, we reduce the number of possible complete assignments of values to variables. Intuitively it follows that this would result in a smaller search space, since the number of possible complete assignments of values to variables provides an upper bound on the size of the search space. Secondly, we show that we can also reduce the number of constraints to be satisfied, since the constraints which were needed to specify that a problem was a sequencing problem in the generic constraint satisfaction representation are now implicitly satisfied in the sequencing representation of the problem.

This report presents the identified problem of basic sequencing problem in job shop sequencing as the constraint satisfaction representation problem. It is a problem in discrete and combinatorial optimization. It is a prominent illustration of a class of problems in computational complexity theory which are classified as NP-hard problem. In this study we explore two models used for multi-level lot-size problem as developed by Billington *et. al.* [1986] and by J.M Wilson [1991] as proposed by Alf Kimms. A comparative study between these models was also given with both models were formulated as an integer-programming problem. A few cases studies were carrying out to show how the system works.

ACKNOWLEDGEMENT

First and foremost, I would like to acknowledge my thanks to Research Management Centre (RMC), Faculty of Science for the support in granting the funds for this project. I also wish to thank my research team Mr Khairil Mahpol, Assoc. Professor Dr Maizah Hura Ahmad and Tan Boon Siew for all the collaboration given. I am extremely grateful to them for giving me advice and many useful suggestions to complete this report.

Finally, I would like to express my heartfelt gratitude to my family and friends for their unstinting support.

ABSTRACT

Lot-size is the clustering of items for transportation or manufacturing processes occurring at the same time. The issue in lot-size problem is to design production processes so that the feasible production quantities are equal to customer demand quantities and the timing of production is such that inventory positions are almost zero. In this study, we explore the multi-level lot-size and scheduling problem. It is on a multi-level capacitated lot-size problem or known as the multi-level lot-size problem with bottlenecks. Two models were introduced to solve the multi-level lot-size problem. They were the model developed by Billington et. al. (1986). It is model with a simple heuristic as proposed by Bilal Toklu and J.M. Wilson (1991), and as proposed by Alf Kimms (1997). A comparative study between these models was also given with both models formulated as an integer-programming problem. Here we represent a study on the implementation of these two models with case studies to demonstrate the systems work.

ABSTRAK

Saiz-lot adalah kelompok item untuk dipindahkan atau kelopmok item dalam proses pengeluaran yang berlaku pada waktu yang sama. Isu dalam masalah saiz-lot bagi rekabentuk proses pengeluaran adalah supaya kuantiti pengeluaran yang tersaur seharusnya sama dengan kuantititi permintaan dan masa pengeluaran di mana kedudukan inventori ada hampir sifar. Dalam kajian ini, kita meneroka masalah saiz-lot berganda dan masalah skedul. Ianya adalah mengenai masalah saiz-lot berkekangan berganda atau lebih dikenali sebagai masalah saiz-lot berganda aras dengan kejajalan. Di sini kita memperkenalkan dua model untuk menyelesaikan masalah saiz-lot berganda. Model ini telah dibangunkan oleh Billington et. al. [1986] dan kaedah huristik mudah oleh Bilal Toklu and J.M. Wilson [1991] bersama model yang diperkenalkan oleh Alf Kimms [1997]. Satu kajian perbandingan terhadap dua model ini telah diberikan sebagai model yang dibangunkan berasaskan masalah pengarturccaraan linear. Di sini saya membentangkan kajian mengimplikasikan kedua-dua model ini dan disertakan dengan beberapa kajian kes bagi menunjukkan bagaimana sistem ini beroperasi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOL

I_{jt}	-	Inventory for item *j* at the end of period *t*
q_{jt}	-	Production quantity for item *j* in period *t*.
x_{jt}	-	Binary variable which indicates whether a setup for item *j* Occurs in period *t* $(x_{jt}=1)$ or not $(x_{jt}=0)$.
y_{jt}	-	Binary variable which indicates whether the machine is set up for item *j* in period *t* $(y_{jt}=1)$ or not $(y_{jt}=0)$.
q_{jn}	-	Production quantity for item *j* at position *n*.
x_{jn}	-	Binary variable which indicates whether a setup for item *j* occurs in position *n* $(x_{jn}=1)$ or not $(x_{jn}=0)$.
y_{jn}	-	Binary variable which indicates whether the machine is set up for item *j* in position *n* $(y_{jn}=1)$ or not $(y_{jn}=0)$.
a_{ji}	-	'Gozinto' factor. Its value is zero if item *i* is not an immediate successor of item *j*. Otherwise, it is the quantity of item *j* that is directly needed to produce one item *i*.
C_{mt}	-	Available capacity of machine *m* in period *t*
CAP_t	-	Available capacity (in time units) of the work centre at time ***t***
d_{jt}	-	External demand for item *j* in period *t*.
h_j	-	Non-negative holding cost for having one unit of item *j* one period in inventory.
I_{j0}	-	Initial inventory for item *j*.
Γ_m	-	Set of all items that share the machine m, i.e. $\Gamma_m \overset{def}{=} \{j \in \{1,\ldots,J\} \mid m_j = m\}$.

Ω	-	Set of end item
J	-	Number of items.
M	-	Number of machines.
	-	The number of bottleneck (Billington el at model)
m_j	-	Machine on which item j is produced.
p_j	-	Capacity needs for producing one unit of item j.
b_j	-	Time needed on the bottleneck facility for the production of product ***j***.
s_j	-	Setup time for the work centre for product ***j***. This takes the value **0** for all items except those made on the work centre. This can also include processing time, which is not related to the size of batch as in some heating operations
cs_j	-	Non-negative setup cost for item j
S_j / $S(j)$	-	Set of immediate successors of item j, *i.e.* $S_j \overset{def}{=} \{i \in \{1,\dots,J\} \mid a_{ji} > 0\}$
T	-	Number of periods
v_j	-	Positive and integral lead time of item j or calls safety lead-time for product ***j***, which is the unavoidable time from the time the order placed until it is available, for product ***j.*** This could be because of time taken by a vendor to deliver a product, or could be a non-production lag
y_{j0}	-	Unique initial setup state.

Single-level lot sizing and scheduling

- Decision variables

Symbol	Definition
I_{jt}	Inventory for item *j* at the end of period *t*.
q_{jt}	Production quantity for item *j* in period *t*.
x_{jt}	Binary variable which indicates whether a setup for item *j* occurs in period *t* $(x_{jt}=1)$ or not $(x_{jt}=0)$.
y_{jt}	Binary variable which indicates whether the machine is set up for item *j* in period *t* $(y_{jt}=1)$ or not $(y_{jt}=0)$.
q_{jn}	Production quantity for item *j* at position *n*.
x_{jn}	Binary variable which indicates whether a setup for item *j* occurs in position *n* $(x_{jn}=1)$ or not $(x_{jn}=0)$.
y_{jn}	Binary variable which indicates whether the machine is set up for item *j* in position *n* $(y_{jn}=1)$ or not $(y_{jn}=0)$.

- Parameters

Symbol	Definition
C_t	Available capacity of the machine in period *t*.
d_{jt}	External demand for item *j* in period *t*.
h_j	Non-negative holding costs for item *j*.
I_{j0}	Initial inventory for item *j*.
J	Number of items.
p_j	Capacity needs for producing one unit of item *j*.
s_j	Non-negative setup costs for item *j*.
T	Number of periods.
N_t	Maximum number of lots in period *t*.

Continuous time lot sizing and scheduling = BSP (Batching and scheduling problem)

- Decision variables

Symbol	Definition
r_n	Completion time of job *n*.
x_{nk}	Binary variable which indicates that job *n* is scheduled right before job *k*.

- Parameters

Symbol	Definition
B	A big number.
f_n	Deadline for job *n*.
h_j	Holding costs for item *j*.
$j(n)$	The item for which job *n* represents demand.
N	Number of jobs.
p_n	Processing time of job *n*.
s_{ji}	Sequence dependent setup costs for items.

Multi-level lot sizing and scheduling

- Parameters

Symbol	Definition
a_{ji}	'Gozinto' factor. Its value is zero if item *i* is not an immediate successor of item *j*. Otherwise, it is the quantity of item *j* that is directly needed to produce one item *i*.
C_{mt}	Available capacity of machine *m* in period *t*.
CAP_t	Available capacity (in time units) of the work center at time ***t***
d_{jt}	External demand for item *j* in period *t*.
h_j	Non-negative holding cost for having one unit of item *j* one period in inventory.
I_{j0}	Initial inventory for item *j*.
Γ_m	Set of all items that share the machine m, i.e. $\Gamma_m \overset{def}{=} \{j \in \{1,...,J\} \mid m_j = m\}$.
Ω	Set of end item
J	Number of items.
M	Number of machines. The number of bottleneck (Billington el at model)
m_j	Machine on which item *j* is produced.
p_j	Capacity needs for producing one unit of item *j*.
b_j	Time needed on the bottleneck facility for the production of product ***j***.
s_j	Setup time for the work centre for product ***j***. This takes the value **0** for all items except those made on the work center. This can also include processing time, which is not related to the size of batch as in some heating operations

cs_j	Non-negative setup cost for item *j*.
S_j / $S(j)$	Set of immediate successors of item *j*, *i.e.* $S_j \overset{def}{=} \{i \in \{1,...,J\} \mid a_{ji} > 0\}$
T	Number of periods.
v_j	Positive and integral lead time of item *j* or calls safety lead-time for product ***j***, which is the unavoidable time from the time the order placed until it is available, for product ***j.*** This could be because of time taken by a vendor to deliver a product, or could be a non-production lag
y_{j0}	Unique initial setup state.

Chapter 1

1.0 Introduction

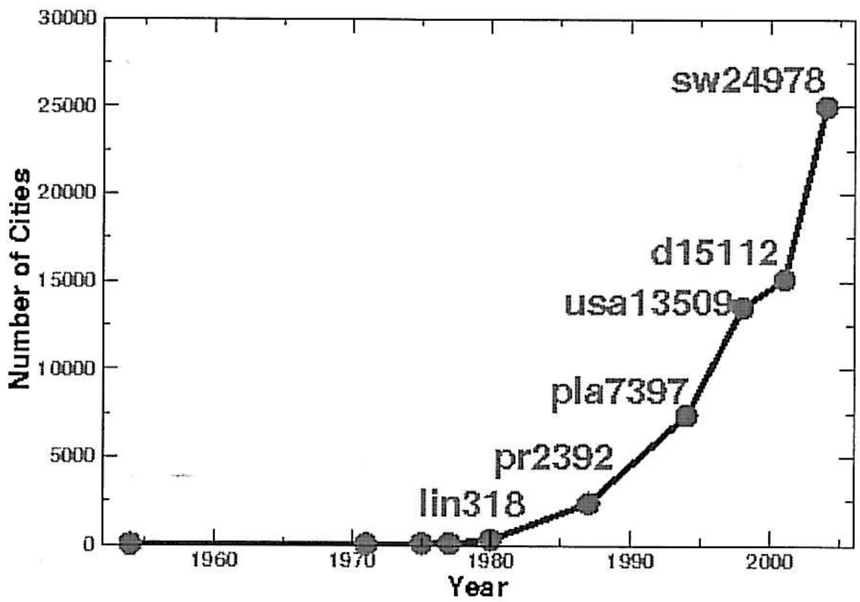
The lot-sizing problem in material requirement planning system is a sequencing problem where the central issue is in the designing of the system in workstations. The multilevel lot-sizing problem is computationally a very difficult to solve. Therefore, it is important to develop effective heuristics for these problems. Heuristics is a way a person learn, discover or solve problems than guessing and acting at random. Therefore, the heuristics method presented here is based on integer programming develop by Billington (1983), when there is a bottleneck facility and at other workstations. A bottleneck is a workstation, which converts raw material into finished goods through the used of the resources in the manufacturing problem. Or in other words, a bottleneck is a work center that limits the production rate of the entire system Billington (1986). The example of bottlenecks is, a machine with limited capacity, highly skilled or specialized workers, and task-specific machines or tools, where, the capacity of the bottleneck is only slightly greater than demand over the horizon or the capacity is exceeded from time to time by demand.

1.1 The Background of the Research Project

1.1.1 Travelling Salesman Problem

How should we measure progress in solving instances of the TSP? A simple judgment is to say that method A is superior to method B if A requires less time or less resources to solve every instance of the problem. This is a clean rule, but it makes direct rankings of methods next to impossible since closely related methods would yield such a simple comparison. It seems necessary that we considerably relax our comparison criterion.

To this end, a judge with a more open mind might be willing to ignore results on very small instances since these can be solved by all good solution techniques. Taking this further, for a given number of cities *n*, the judge might want to concentrate on those *n*-city instances that cause the most difficulty for a proposed method that he or she must evaluate. Adopting this approach, we would rank method A ahead of method B if for every large value of *n* the worst *n*-city example for A takes less time to solve than does the worst *n*-city example for B. The fundamental aspect of TSP is the mathematical formulation of the classical problem. To make this comparison idea work in practice, we can analyze a given solution method to obtain a a guarantee that it takes at most some amount of time f(*n*) for any *n*-city TSP, where f(*n*) is shorthand for some formula that depends only on *n*. Now to compare two solution methods, we compare the best guarantees that we have found for them. This may of course produce misleading results since a really good method might just be tough to analyze and therefore appear to be poor when compared to a method that leads to a good analysis. On many computational problems, however, the study of algorithms and guarantees has led to some beautiful mathematical results as well as important improvements in practical problems.



Solving an instance of the travelling salesman problem (TSP) does not mean finding a good tour or even finding one that is better than any previously known. To solve a TSP we need to find an absolute shortest tour and to know indeed that no better tour is possible. Given the complexity of the calculations it seems unlikely that we will ever be able to construct a proof that our Sweden tour is optimal that can be checked completely by hand, without the aid of a computer. Indeed, our Sweden TSP computation used 84.8 CPU years to verify that a best-possible tour had been found. Our verification process included numerous checks to ensure that the computation was accurate. Areas for exploration include the Geometric lower bounds with control zones , Sub-tour elimination, The cutting-plane method , Local cuts, Branch-and-cut and Accuracy of computations

1.1.2 Research Problem

A simple problem for lot sizing cases can be formulated for a single stage with infinite production capacity and a single product to be planned over T periods of time:

Minimize $\sum_{t=1}^{T} hI_t$

Subject to:

$$I_{t-1} + Q_t - I_t = d_t, \qquad \forall t = 1,...,T$$

$$I_t \geq 0 \text{ and } x_t \geq 0 \qquad \forall t = 1,...,T$$

Where *h* represent the inventory holding cost of the product from one period to the next and d_t represents the product demand at the end of the period *t*. Then the Q_t represent the quantity to be produced at in that particular period *t* and the I_t represents the inventory of the product at the end of period *t*.

A general product structure with two-bottleneck facility is given as Figure 1, which is given by Billington (1986).

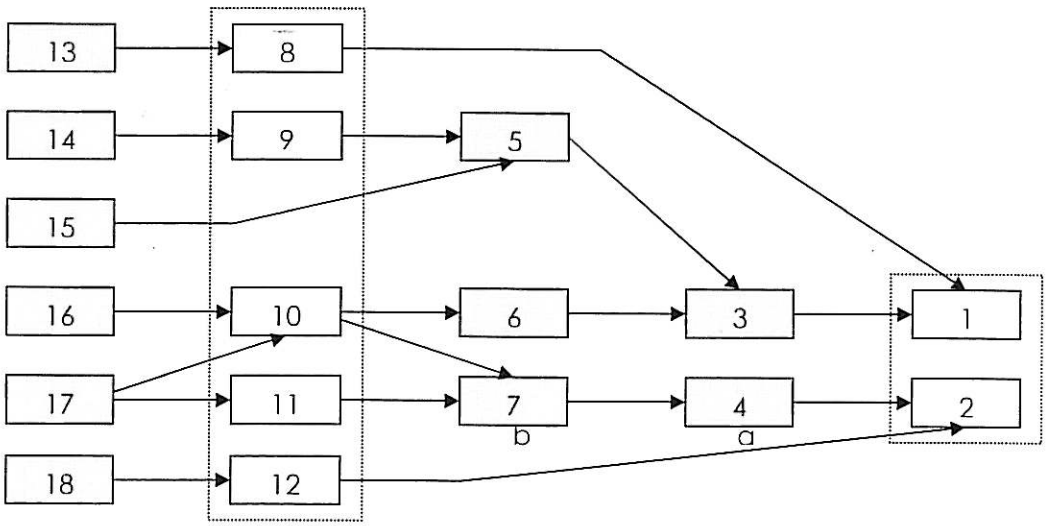


Figure1: A general product structure with multiple bottleneck facility (the bottleneck facility is shown as the dashed lines.

The general product structure can be split into special cases; such as:

(1) assembly (no commonality),

(2) serial (one item, multi-stage),

(3) parallel (a combination of serial structure with a bottleneck facility in one of the stages)

(4) single stage multi-item.

However, we will only look at a few cases in this paper. From the Figure above, there is no item with a higher number than any of it's predecessors (b>a). Here Billington is assumed that the items in the bottleneck facility do not have predecessors.

Lot sizing is one of the critical decisions made during the tactical planning stage of a manufacturing process. It concerns with determination of the production quantities and production timing of products to meet demand requirements over the planning horizon. Due to limited resources in the process to decide how much they can produce in a period of production time. This is a dilemma for a production engineer in selecting a procedure. Therefore, this study attempt to formulate heuristic method to produce an effective model where it will choose the lot sizes, which minimize the total of setup cost and the inventory holding cost.

1.2 Objective of the Study

This study has two separate objectives as follows

- Modelling real-world job shop sequencing problem
- Formulate heuristic method to produce an effective model for choosing the lot sizes, which minimize costs.

1.3 Organisation of The Report

In this report, I have been divided it into five chapters.

In the first chapter, there is an introduction about what is Job Shop scheduling, the lot-sizing problem with bottleneck, with a simple example of lot-size problem. Furthermore, the over view about the objective of this study.

More over, the review of the literature about the heuristic for multilevel lot sizing problem with a bottleneck, will discuses in the chapter 2. Furthermore, we will discuses about some general lot sizes problem.

In the chapter 3, the method use to solve the lot-sizing problem with bottleneck will be introduced. Thus, the integer programming with heuristic will be discusses in this chapter. Some information about the LINDO software will also introduce in this chapter since we choose this software to help to get the solution of our problem.

Then, the cases study and the result will shown in the chapter 4 where, three case of simple lot sizing problem with bottleneck will discusses in this chapter.

Finally, in chapter 5 consists the conclusions and discussion.

Chapter 2

Literature Review

2.0 Introduction

Lot sizing is the clustering of items for transportation or manufacturing processing at the same time. Lot- sizing also called batching. It is a mechanism that induces time-phased production that is usually nonsynchronized with the actual assumption or demand pattern. As we have mention earlier, this research is concentrates on multilevel lot sizing problems with bottleneck under rolling schedule environment and will focus on using a simple heuristic to solve the problem. In this chapter, we will discuss the general lot-sizing problem, the methods used and the problem insights.

2.1 The general Lot-sizing problem

Capacity supply is usually modelled as a function of time, stating the number of resource units available per unit of time. The main problem for many firms is to decide how much they can produce with limited resources. Lot- sizing decisions are taken with respect to flow and storage of material and information. It combines the requirements and the production orders in the planning horizon. Furthermore, lot-sizing rules do not provide the correct period for placing the requirements but they determine the quantity of orders for a part or finished product is given by the lot-sizing rules.

The basis in lot sizing is the trade off between inventory and setups under consideration of the cost, service effectiveness and most of all, the holding cost. Holding cost are the cost of holding item in storage and a setup cost is the cost which is not depend on the order quantity is incurred whenever an order is placed during the planning horizon. Therefore, the objective of this research is to choose the lot-sizes, which will minimize the total setup and inventory holding costs.

2.1.1 The Structure / Systems

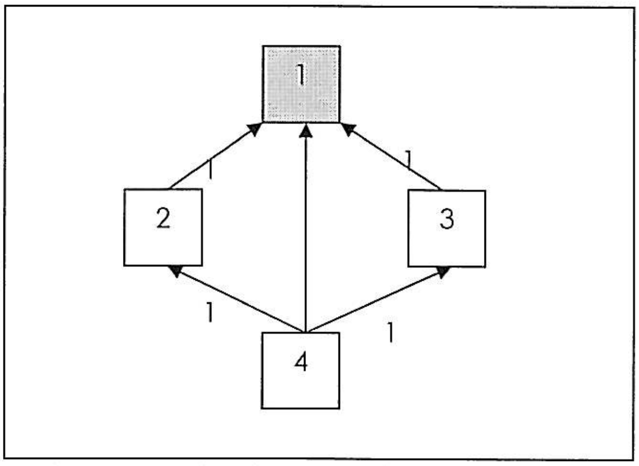


Figure 2 : A simple general product structure

Figure 2 shows the simple general product structure, it is also call Gozinto-Structure or also known as acyclic graph (directed network). It gives a general idea about what is the structure of a product. It is compact and easy to read, but it misleadingly reflect the precedence relations for scheduling items. There is another extension graph, which is also give us the same information, call Gozinto-tree (figure 3). It represented a directly reveals the precedence relations in a feasible schedule which gained by converting the general Gozinto-graph into an assembly structure by copying nodes with more than one successor. The detail about Gozinto-graph has been discussed by Alf Kimms [1].

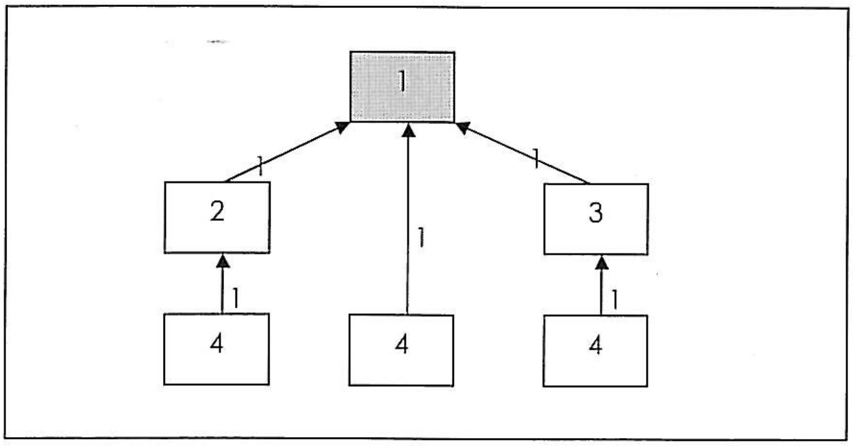


Figure 3 : Gozinto-tree

Incidentally, Figure 4 shows a multi-level/stage inventory system, which it is a connected set of stage representing the steps for assembly and/or distribution for a family of products. As describe earlier, the general product structure can be split into the following special cases;

(1) Assembly,
(2) Serial,
(3) Parallel,
(4) Single stage multi-item.
(Example of such systems are given in figure 4)

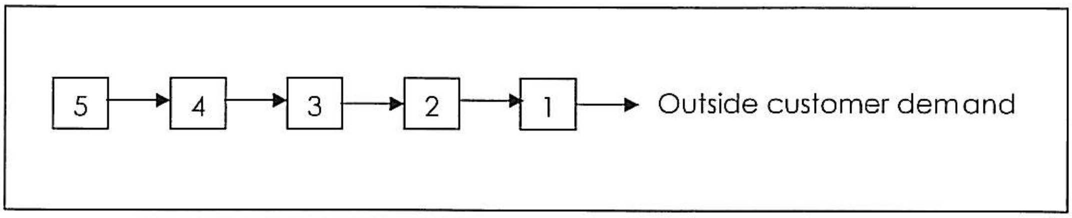


Figure 4 (a): 5 stage serial systems

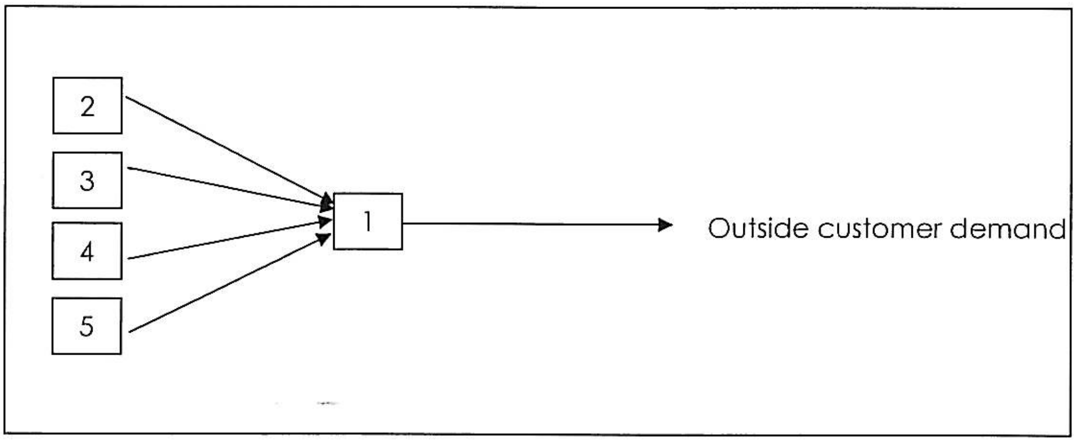


Figure 4 (b): 5 stage assembly system

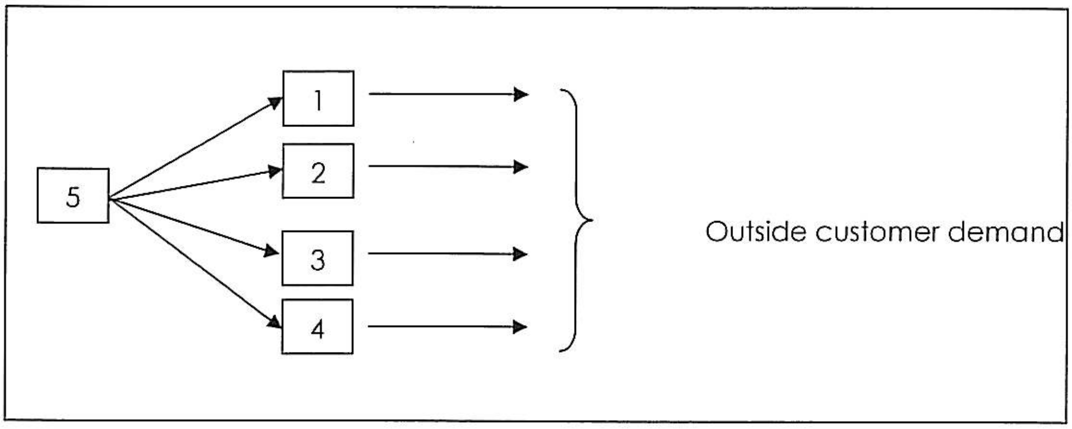


Figure 4 (c): 5 stage distribution system

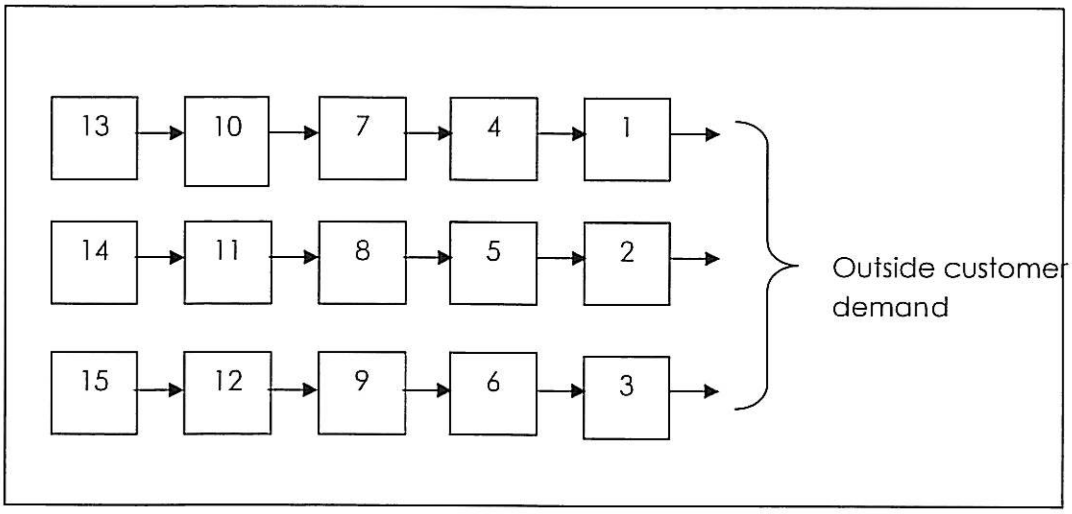


Figure 4 (d): 15 stage parallel system

A standard classification is to differentiate between assembly and distribution inventory systems. Fig 4 (b) shows an assembly system where each stage has at most one immediate successor, while Fig 4 (c) shows a distribution system where each stage has at most one immediate predecessor. Fig 4 (a) shows a serial system, where it is both distribution and assembly system, and then the Fig 4 (d) is a structure called parallel structure, which is the collection of serial system. While a general multiple stage system need be neither a distribution nor assembly system.

A survey has conduct by A. Drexl, and Alf Kimms [6] in the field of Lot-sizing and scheduling. They discuss about the capacitated lot-sizing problem (CLSP), the discrete lot-sizing problem (DLSP), the continuous setup lot-sizing problem (CSLP) and the proportional lot-sizing problem (PLSP). However, they also discuss detail on the continuous time model and multi level lot sizing and scheduling problem. They gave a clear direction and steps on how to solve the production-planning problem. Additionally, they have list out a stepwise manner to construct feasible production plans. The phases are outlined as follows:

Phase I: starting with end items, lot sizes are computed level by level for all items in the multi-level gozinto structure. By doing so, the capacity constraints are ignored.

Phase II: the result obtained by phase I usually exceeds the available capacity in some period. Hence, some lots are shifted in order to find a plan, which meets the capacity limits. By doing so, the precedence relations among the items are ignored.

Phase III: Sequence decisions are made and orders are released to the shop floor.

For this they have considered an example in [6]. Alf Kimms [1] has given a set of clear information about many methods to solve this kind of problems. He describes many methods for capacitated, Dynamics, and Deterministic model and the general problem outline and also the general issue that always happen in the real situation. He also describes the methods to derive PLSP to CSLP, DLSP and also CLSP. Genetic Algorithms (GA) and Taboo search were two of the modern heuristic methods he discussed to solve the MLP. He also discusses about the genetic Algorithms for multi-level, multi machine lot sizing and scheduling in [11]. Furthermore, for whose are interested in solving the MLP by using the GA method. This methods also given in [8], [10], [13] and [15].

2.1.2 CLSP (Capacitated lot sizing problem)

This model is an extension of the Wagner-Whitin (WW) problem to capacity constraints. CLSP is called large bucket problem [36] because several items may be produced per period

Model

$$\text{Min} \sum_{j=1}^{J} \sum_{t=1}^{T} (s_j x_{jt} + h_j I_{jt}) \qquad (1)$$

Subject to

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt},$$
$$j = 1,...,J, \quad t = 1,...,T, \quad (2)$$

$$p_j q_{jt} \le C_t x_{jt}, \quad j = 1,...,J, \quad t = 1,...,T, \quad (3)$$

$$\sum_{j=1}^{J} p_j q_{jt} \le C_t, \quad t = 1,...,T, \quad (4)$$

$$x_{jt} \in \{0,1\}, \quad j = 1,...,J, \quad t = 1,...,T, \quad (5)$$

$$I_{jt}, q_{jt} \ge 0, \quad j = 1,...,J, \quad t = 1,...,T, \quad (6)$$

2.1.3 DLSP (Discrete lot sizing and scheduling problem)

DLSP, is a problem, which subdividing the (macro-) periods of the CLSP into several (micro-) periods leads to the discrete lot sizing and scheduling problem. Different between DLSP with CLSP are: (1) There is only one item may be produced per period ('all-or-nothing' production) and if so, production uses full capacity. (2) Setup cost incurred only if the production of a 'new' lot begins. (3) The advantage over CLSP is that minimum lead times, such as transportation time or time for cooling, can easily be taken into account. If the CLSP is used as a basis with periods representing weeks, (short) minimum lead times must either be ignored or be overestimated. The latter leads to high total lead times.

Model

$$\text{Min} \sum_{j=1}^{J} \sum_{t=1}^{T} (s_j x_{jt} + h_j I_{jt}) \quad (7)$$

Subject to

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{8}$$

$$p_j q_{jt} = C_t x_{jt}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{9}$$

$$\sum_{j=1}^{J} y_{jt} \leq 1, \quad t = 1,...,T, \tag{10}$$

$$x_{jt} \geq y_{jt} - y_{j(t-1)}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{11}$$

$$y_{jt} \in \{0,1\}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{12}$$

$$I_{jt}, q_{jt}, x_{jt} \geq 0, \quad j = 1,...,J, \quad t = 1,...,T, \tag{13}$$

2.1.4 CSLP (Continuous setup lot sizing problem)

In CSLP the different between this model with DSLP are there are non-production periods $(t+1,...,t^{'}-1)$ between production period of item j $(t,t^{'})$, and setup is not incurred at $t^{'}$. Furthermore, there are 'all-or-nothing' assumption is given up.

Model:

$$\text{Min} \sum_{j=1}^{J} \sum_{t=1}^{T} (s_j x_{jt} + h_j I_{jt}) \tag{14}$$

Subject to

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{15}$$

$$p_j q_{jt} \leq C_t x_{jt}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{16}$$

$$\sum_{j=1}^{J} y_{jt} \leq 1, \quad t = 1,...,T, \tag{17}$$

$$x_{jt} \geq y_{jt} - y_{j(t-1)}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{18}$$

$$y_{jt} \in \{0,1\}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{19}$$

$$I_{jt}, q_{jt}, x_{jt} \geq 0, \quad j = 1,...,J, \quad t = 1,...,T, \tag{20}$$

2.1.5 PLSP (Proportional lot sizing and scheduling problem)

The basic idea of PLSP is to use remaining capacity for scheduling a second item in the particular period.(cf. CLSP, PLSP, CSLP produce at most 1 item per period.)

The difference between PLSP with CLSP are (1) (at most) 2 items may be produced per period. (use remaining capacity of PLSP model for scheduling a second item in the particular period). (2) setup is need when second item is produced during a period. But, similar to the CSLP, idle periods between two lots of the same item do not cause additional setup costs. (3) If two items are produced in a period, the order for each items are to be produced must be clear. This is accomplished by interpreting the setup state decision variables y_{jt} in the following manner: y_{jt} is the setup state of the machine at the end of a period.

Model:

$$\text{Min } \sum_{j=1}^{J} \sum_{t=1}^{T} (s_j x_{jt} + h_j I_{jt}) \tag{21}$$

Subject to

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt}, \quad j = 1,\dots,J, \quad t = 1,\dots,T, \tag{22}$$

$$p_j q_{jt} \le C_t (y_{j(t-1)} + y_{jt}), \quad j = 1,\dots,J, \quad t = 1,\dots,T, \tag{23}$$

$$\sum_{j=1}^{J} p_j q_{jt} \le C_t, \quad t = 1,\dots,T, \tag{24}$$

$$\sum_{j=1}^{J} y_{jt} \le 1, \quad t = 1,\dots,T, \tag{25}$$

$$x_{jt} \ge y_{jt} - y_{j(t-1)}, \quad j = 1,\dots,J, \quad t = 1,\dots,T, \tag{26}$$

$$y_{jt} \in \{0,1\}, \quad j = 1,\dots,J, \quad t = 1,\dots,T, \tag{27}$$

$$I_{jt}, q_{jt}, x_{jt} \ge 0, \quad j = 1,\dots,J, \quad t = 1,\dots,T, \tag{28}$$

2.1.6 GLSP (General lot sizing and scheduling problem)

GLSP is a lot sizing problem with stationary demand, where each lot is uniquely assigned to a position number in order to define a sequence. The fundamental assumption for the GLSP is that a user-defined parameter restricts the number of lots per period. Let say, when $N_t = 1$ for all $t = 1,...,T$, then the GLSP equals the CSLP.

The difference with DLSP is that GLSP have (1) stationary demand, N_t (Maximum number of lots during period t) is given. (2) A particular item may be produced at several position in a period. The assumption about setup is similar with CSLP.

Model:

$$\text{Min} \sum_{j=1}^{J} \sum_{t=1}^{N} s_j x_{jn} + \sum_{j=1}^{J} \sum_{t=1}^{T} h_j I_{jt} \tag{29}$$

Subject to

$$I_{jt} = I_{j(t-1)} + \sum_{n=F_t}^{L_t} q_{jn} - d_{jt},$$
$$j = 1,...,J, \quad t = 1,...,T, \tag{30}$$
$$p_j q_{jn} \le C_t y_{jn}, \quad j = 1,...,J, \tag{31}$$
$$\sum_{j=1}^{J} \sum_{n=F_t}^{L_t} p_j q_{jn} \le C_t, \quad t = 1,...,T, \tag{32}$$
$$\sum_{j=1}^{J} y_{jt} \le 1, \quad n = 1,...,N, \tag{33}$$
$$x_{jn} \ge y_{jn} - y_{j(n-1)}, \quad j = 1,...,J, \quad n = 1,...,N,$$
$$y_{jn} \in \{0,1\}, \quad j = 1,...,J, \quad n = 1,...,N, \tag{34}$$
$$I_{jt} \ge 0, \quad j = 1,...,J, \quad t = 1,...,T, \tag{35}$$
$$q_{jn}, x_{jn} \ge 0, \quad j = 1,...,J, \quad n = 1,...,N, \tag{36}$$
(37)

2.1.7 Continuous time lot sizing and scheduling = BSP (Batching and scheduling problem)

In this model, each demand is characterized by its deadline and its size. Demands are interpreted as jobs and the demand size determines the processing time of a job. And several demands (=jobs) for the same item may be grouped together to form one lot and to save setup cost. Due to this assumption, the problem is referred to as a batching and scheduling problem (BSP) rather than a lot sizing and scheduling problem.

For this model there are an important assumption, which is that the capacity (e.g. the speed of the machine) is constant over time. And the jobs are not allowed to split.

Model:

$$\text{Min} \sum_{n=0}^{N} \sum_{k=1, k \neq n}^{N} s_{j(n)j(k)} x_{nk} + \sum_{j=n}^{N} h_{j(n)} p_n (f_n - r_n) \quad (38)$$

Subject to

$$\sum_{k=1, k \neq n}^{N+1} x_{nk} = 1, \quad n = 0, \ldots, N, \quad (39)$$

$$\sum_{k=1, k \neq n}^{N} x_{kn} = 1, \quad n = 1, \ldots, N+1, \quad (40)$$

$$r_n + p_k \leq r_k + B(1 - x_{nk}), \quad n = 0, \ldots, N, \; k = 1, \ldots, N+1, \quad (41)$$

$$r_n \leq f_n, \quad n = 1, \ldots, N, \quad (42)$$

$$x_{nk} \in \{0,1\}, \quad n = 0, \ldots, N, \quad k = 1, \ldots, N+1, \quad (43)$$

$$r_n \geq 0, \quad n = 1, \ldots, N+1, \quad r_0 = 0. \quad (44)$$

2.2 Multi-level lot sizing and scheduling

Basically, the model is similarly with the single-level problem. However, there still have some different between these problems. Here are the differences and similarities of this model with the single level problem. (1) Sequence dependencies and lead times can be incorporated into the model. (2) Idle periods among jobs for the same item do not cause additional setups, which are similar to the CSLP, the PLSP, and the GLSP. (3) (multi-level) DLSP and the (multi-level) CSLP are special cases of the (multi-level) PLSP.

The model below shows us the Multi-level PLSP.

Models:

$$\text{Min} \sum_{j=1}^{J} \sum_{t=1}^{T} (s_j x_{jt} + h_j I_{jt}) \tag{45}$$

Subject to

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} - \sum_{i \in S_j} a_{ji} q_{it}, \tag{46}$$

$$j = 1,...,J, \quad t = 1,...,T,$$

$$I_{jt} \geq \sum_{j \in S_j} \sum_{\tau = t+1}^{\min\{t+v_j, T\}} a_{ji} q_{i\tau}, \quad j = 1,...,J, \quad t = 0,...,T-1, \tag{47}$$

$$p_j q_{jr} \leq C_{m_j t}(y_{j(t-1)} + y_{jt}), \quad j = 1,...,J, \quad t = 1,...,T, \tag{48}$$

$$\sum_{j \in \Gamma_m} p_j q_{jt} \leq C_{mt}, \quad t = 1,...,T, \quad m = 1,...,M, \tag{49}$$

$$\sum_{j \in \Gamma_m} y_{jt} \leq 1, \quad m = 1,...,M, \quad t = 1,...,T, \tag{50}$$

$$x_{jt} \geq y_{jt} - y_{j(t-1)}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{51}$$

$$y_{jt} \in \{0,1\}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{52}$$

$$I_{jt}, q_{jt}, x_{jt} \geq 0, \quad j = 1,...,J, \quad t = 1,...,T, \tag{53}$$

For those people, whose are very interested in how to derive the model above into multilevel CSLP, multilevel DLSP and also multilevel CLSP, have clearly describe by Alf Kimms [1]. Andréa Toniolo Staggemeier and Alistair R. Clark also do the same thing as A. Drexl, and Alf Kimms did. The different is they concentrate more in lot-sizing and scheduling models for single-stage problem. The propose of this paper is present different aspects of such models in the operation research area and notes the most common modern methods to solve them. Which are the Metaheuristic methods, such as, Taboo Search, Simulated Annealing (SA), Genetic Algorithms, Integer Programming and so on. They considered several important aspects, such as, capacity constraints, backlogs, setup costs, multistage problem, multiple item problem, and size of planning period. The paper started with a very simple problem gradually including these aspects and variations of the problem in a structured manner to show the major features found in Lot-Sizing and Scheduling problems. Moreover, they classified a referenced list of modern solution methods for multiple-product and multiple-machine lot sizing and scheduling problems.

2.2.1 The serial lot-sizing systems

A serial system is the basic form of the multi-level lot-sizing problem, that is the most simplified form, involves producing one product in multi-level/stage incapacitated production process. The multi-level characteristic is derived from the hierarchical nature of the manufacturing process where raw materials are processed into components, components into subassemblies, and so on until the final product is completed. (Figure 3(a)). Stephen F. Love [1972] consider a production planning in which N facilities are arranged in series with constant production cost and storage cost. This model is a multi-echelon/level structure. In their research, no backlogging are allowed and he assume that one unit of of production at any facility requires input of one unit item from preceding facility. Furthermore, it also assume that no overproduction takes place and the lead-times of the preceding item are equal to zero, or in other words, the production is instantaneous. They have show that if storage and production costs are nondecreasing in order of facilities and nonincreasing in time, then the optimal schedule has the property that in a same period facility j and j+1 can produce together. They have developed an algorithm to find an optimal schedule for this nested structure.

Billington el at. [1986] consider capacity constrained multi-level parallel scheduling problem with a bottleneck. Their formulation is based on Billington el at. [1983], except that only one capacity-constrained work centre exist. They eliminated inventory by substituting cumulative production minus cumulative demand rather than using both inventory (I_{it}) and production (q_{jt}) variables. The formulation are given as below:

Minimize:

$$Z = \sum_{j=1}^{J} \sum_{t=1}^{T} [h_j (T-t+1) q_{jt} + cs_j X_{jt}] \tag{54}$$

Subject to:

$$\sum_{n=1}^{t} q_{j,n-v_j} - \sum_{i=1}^{t} a_{ji} q_{in}] \geq \sum_{n=1}^{t} d_{jn} - I_{j^\circ} \qquad j=1,2,\ldots,J \quad t=1,2,\ldots,T \qquad (55)$$

$$\sum_{j=1}^{J} [b_j q_{jt} + s_j X_{jt}] \leq CAP_t \qquad t=1,2,\ldots,T \qquad (56)$$

$$X_{jt} = \begin{cases} 1; & \text{if } q_{jt} > 0 \\ 0; & \text{otherwise} \end{cases} \qquad j=1,2,\ldots,J \qquad (57)$$

$$q_{jt} \geq 0 \qquad j=1,2,\ldots,J \quad t=1,2,\ldots,T \qquad (58)$$

They solve the problem by using branch and bound with heuristic. Where they use Lagrangian relaxation method to embed branch and bound procedure. In their paper, they have shown us many example of problem. They separate the problem into special cases and the general problem.

Shawnee K. Vickery and Robert E. Markland studied the development and implementation of a large-scale, multi-objective, lot-sizing model for scheduling tablet pharmaceuticals in a serial production system. They concentrate in the scarcity of multi-stage lot-sizing models that incorporate capacity restrictions into the lot-sizing process. Where, they developing a multi-item, multi-stage lot-sizing procedure for serial production systems characterized by one or more multi- resource capacitated stages. The test problems in their paper are similar in nature to the 'capacity/lot-size/lead-time that was defined by Billington el at [1983] for the general product structure. The difference is Shawnee and Robert uses historical lead-time data to develop a multi-stage lot-sizing model while Billington el at viewed lead-time and capacity utilization as outputs of their MRP-ILP.

2.2.2 Non-serial lot-sizing systems

Generally, there are two systems, which are considering as non-serial lot-sizing systems. There are assembly system and the general production structure. Where, assembly structure may have only one successor and more than one predecessor; while general product structure have more that one successor and predecessor.

Horst Tempelmeier el at [1993] studied a dynamic lot-sizing problem in general non-cyclic product/operation structures under multiple capacity constraints. They modified the Dixon-Silver heuristic to solve a sequence of single-level capacitated multi-item lot-sizing problems. Here, they consider two costing methods and two ways of ordering the product/operation structure. The tested problem using by Horst and his friends is the problem, which the optimal solution is available. They have list down two major drawbacks for this planning strategy:

a) Lot-sizes are computed level by level with *a single-stage lot-sizing model* without consideration of multi-level interaction effects between lot-sing decisions. All lot-sizing decisions on a given parent level are made before proceeding to the next level (component). Thus, at best, a sub optimazation is achieved

b) Lot-sizes are computed under the assumption of *infinite capacity.* As a result, often the computed lot-sizes are infeasible with respect to the capacity of the bottleneck resources. As a result, the production planner change the lot-sizes manually due to limited human problem solving capacity, unpredictable lead times and sub optimal production plans to generate a feasible production schedule.

Horst el at concentred in two type of lot-sizing problem, which is the single-level, capacitated lot-sizing problem (CLSP) and multi-level, uncapacitated lot-sizing problem (MLUSP). For the CLSP, the formulation using in this paper are devise from Dixon and silver [1981]. Afterwards, the formulations of MLUSP for general product structure are devising by other authors such as Heinrich, Jacobs and Khumawala, Mclaren and so on. The formulation of Multi-Level, Capacitated Lot-Sizing Problem (MLCLSP) given by Horst el at is shows as follows:

Model MLCLSP

Minimize:

$$Z = \sum_{j=1}^{J} \sum_{t=1}^{T} [h_j I_{jt} + cs_j X_{jt}] \qquad (59)$$

Subject to:

$$I_{j,t-1} + q_{j,t-v_j} - \sum_{i \in S(j)} a_{ji} q_{it} - I_{jt} = d_{jt} \qquad \begin{array}{l} j = 1,2,\ldots,J \\ t = 1,2,\ldots,T \end{array} \qquad (60)$$

$$\sum_{j \in \Gamma_m} [b_{jm} q_{jt}] \le C_{mt} \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (61)$$

$$X_{jt} = \begin{cases} 1; & \text{if } q_{jt} > 0 \\ 0; & \text{otherwise} \end{cases} \qquad j = 1,2,\ldots,J \qquad (62)$$

$$q_{jt}, I_{jt} \ge 0 \qquad \begin{array}{l} j = 1,2,\ldots,J \\ t = 1,2,\ldots,T \end{array} \qquad (63)$$

Then to calculated the total exploded requirements for item *j* in the time period *t*, including possible external demand for item *j* are given by

$$D_{jt} = d_{jt} + \sum_{i \in \Omega} \rho_{ji} d_{it} \qquad \begin{array}{l} j = 1,2,\ldots,J \\ t = 1,2,\ldots,T \end{array} \qquad (64)$$

$$D_{jt\tau} = \sum_{l=t}^{\tau} D_{jl} \qquad \begin{array}{l} j = 1,2,\ldots,j \\ t = 1,2,\ldots,T \\ \tau = t, t+1,\ldots,T \end{array} \qquad (65)$$

Where (7) calculated the requirements of item *j* for period t to τ. As ρ_{ji} is the total requirements of item j need to product one unit of item i.

Then the Multi-level Uncapacitated Lot-sizing Problem (MLULSP),is given as:

$$\text{Minimize } z = \sum_{j=1}^{J} \frac{cs_j}{t_j} + (t_j - 1).D_j.\frac{1}{2}e_j \tag{67}$$

Subject to:

$$t_j \geq t_i \qquad \begin{matrix} j = 1,2,\ldots,J \\ i \in S(j) \end{matrix} \tag{68}$$

$$t_j \in \{1,2,4,8,16,\ldots\} \qquad j = 1,2,\ldots,J \tag{69}$$

Where D_j, cs_j, e_j and t_j are the mean total demand, the setup cost, the echelon holding cost and the length of the production cycle for item *j* respectively. Chia-Shin Chung, James Flynn and Chien-Hua Mike Lim, [1994] studies a deterministic, single product capacitated dynamic lot size model with linear production and holding costs where the setup costs, unit production costs and capacities are arbitrary functions of the period and the unit production costs satisfy the growth constraints which is the unit production cost in any period can never exceed the sum of the unit production cost and the unit holding cost in previous period. Where, they have developed an algorithm, which combines dynamic programming with branch and bound method. R.Kuik and M.Salomon studies in multi-level lot-sizing problem with solve the problem by using simulated-annealing (SA) heuristics. They investigate heuristics based on a stochastic search method. In their paper, they make comparison of the test problem between SA and MRP level-by-level heuristics. Though their result, SA method given better solution than MRP level-by level heuristic for the large problems. Even, SA heuristic given better solution, but it needed/ takes large computation time. Though, R.Kuik and M.Salomon another disadvantage also exists, that is we no know how far a solution given differ from optimality. However, this paper has given us a general /overview about how to solve a MRP problem by using SA method. they have shows us the implementation of these heuristic and also the model formulation. B. Toklu and J.M. Wilson examine a simple heuristic for multilevel lot-sizing problem with multiple bottlenecks. They suggest that items to be produce can be grouped into two types, which is:

(a) Items constrained by the bottleneck.

(b) Items, which are unconstrained.

They have developed two simple procedures, which can used independently, one for each category of item, to determine the production levels of each item. First of all, they were dividing production items into end-items and non-end-items. Where, either some of the non-end items are constraint by the bottleneck or both of them. In their studies, they consider in the problem, which only have two bottlenecks. Moreover, one of the bottlenecks is always taken to occur in the end-items and the second one could be any other intermediate or row material stage, in their case call non-end-items stage. In the paper they discusses about the non-end items outside bottleneck, non-wend items with bottleneck, and the end items with bottleneck. The formulation use in this paper is derived from a model in Billington el at. [1986]. The only difference is now the parameter of times need on the bottleneck facility for the production of production i is become b_{jl}, where *l* is the number of bottleneck facility. The detail and the formulation will be given in chapter 3.

2.3 Tabu Search and Its relevance

Faced with the challenge of solving hard optimization problems that abound in the real world,classical methods often encounter great difficulty. Vitally important applications in business, engineering, economics and science cannot be tackled with any reasonable hope of success, within practical time horizons, by solution methods that have been the predominant focus of academic research throughout the past three decades (and which are still the focus of many textbooks).

The meta-heuristic approach called Tabu Search (TS) is dramatically changing our ability to solve problems of practical significance. Current applications of TS span the realms of resource planning, telecommunications, VLSI design, financial analysis, scheduling, space planning, energy distribution, molecular engineering, logistics, pattern classification, flexible manufacturing, waste management, mineral exploration, biomedical analysis, environmental conservation and scores of others. In recent years, journals in a wide variety of fields have published tutorial articles and computational studies documenting successes by Tabu Search in extending the frontier of problems that can be handled effectively — yielding solutions whose quality often significantly surpasses that obtained by methods previously applied. Table 1.1 gives a partial catalog of example applications. A more comprehensive list, including summary descriptions of gains achieved from practical implementations, can be found in Glover and Laguna, 1997. Reports of recent TS implementations can also be found on the web page http://www.upt.pt/tabusearch.

2.4 Review on Applications of Tabu Search

A distinguishing feature of Tabu Search is embodied in its exploitation of adaptive forms of memory, which equips it to penetrate complexities that often confound alternative approaches. Yet we are only beginning to tap the rich potential of adaptive memory strategies, and the discoveries that lie ahead promise to be as important and exciting as those made to date. The knowledge and principles that have emerged from the TS framework give a foundation to create practical systems whose capabilities markedly exceed those available earlier. At the same time, there are many untried variations that may lead to further advances. A conspicuous feature of Tabu Search is that it is dynamically growing and evolving, drawing on important contributions.

Table 1. Illustrative Tabu Search Applications.

Scheduling and Telecomunication	Design
Scheduling	Computer-Aided Design
Flow-Time Cell Manufacturing	Fault Tolerant Networks
Heterogeneous Processor Scheduling	Transport Network Design
Workforce Planning	Architectural Space Planning
Classroom Scheduling	Diagram Coherency
Machine Scheduling	Fixed Charge Network Design
Flow Shop Scheduling	Irregular Cutting Problems
Job Shop Scheduling	
Sequencing and Batching	
Telecommunications	
Call Routing	
Bandwidth Packing	
Hub Facility Location	
Path Assignment	
Network Design for Services	
Customer Discount Planning	
Failure Immune Architecture	
Synchronous Optical Networks	

Production, Inventory and Investment	Routing
Flexible Manufacturing Just-in-Time Production Capacitated MRP Part Selection Multi-item Inventory Planning Volume Discount Acquisition Fixed Mix Investment **Location and Allocation** Multicommodity Location/Allocation Quadratic Assignment Quadratic Semi-Assignment Multilevel Generalized Assignment Lay-Out Planning Off-Shore Oil Exploration	Vehicle Routing Capacitated Routing Time Window Routing Multi-Mode Routing Mixed Fleet Routing Travelling Salesman Travelling Purchaser
Logic and Artificial Intelligence	Graph Optimization, technology and General Optimization Problem
Maximum Satisfiability Probabilistic Logic Clustering Pattern Recognition/Classification Data Integrity Neural Network |Training and Design	**Graph Optimization** Graph Partitioning Graph Colouring Clique Partitioning Maximum Clique Problems Maximum Planner Graphs P-Median Problems **Technology** Seismic Inversion Electrical Power Distribution Engineering Structural Design Minimum Volume Ellipsoids Space Station Construction Circuit Cell Placement **General Combinational Optimization** Zero-One Programming Fixed Charge Optimization Non-convex Nonlinear Programming All-or-None Networks Bi-level Programming General Mixed Integer Optimization

2.5 Summary

Fred Glover in 1986 introduced the basic concept of Tabu Search as "a meta-heuristic superimposed on another heuristic. The basic idea of this approach is to avoid getting stranded in cycles by forbidding or penalizing moves which take the solution, in the next iteration, to points in the solution space previously visited (hence "tabu"). The Tabu search method is fairly new as the method is still actively researched, and is continuing to evolve and improve. The Tabu method was partly motivated by the observation that human behavior appears to operate with a random element that leads to inconsistent behavior given similar circumstances. As Glover points out, the resulting tendency to deviate from a charted course, might be regretted as a source of error but can also prove to be source of gain. The Tabu method operates in this way with the exception that new courses are not chosen randomly. Instead the Tabu search proceeds according to the supposition that there is no point in accepting a new (poor) solution unless it is to avoid a path already investigated. This insures new regions of a problems solution space will be investigated in with the goal of avoiding local minima and ultimately finding the desired solution.

The Tabu search begins by marching to a local minima. To avoid retracing the steps used, the method records recent moves in one or more Tabu lists. The original intent of the list was not to prevent a previous move from being repeated, but rather to insure it was not reversed. The Tabu lists are historical in nature and form the Tabu search memory. The role of the memory can change as the algorithm proceeds. At initialization the goal is make a coarse examination of the solution space, known as 'diversification', but as candidate locations are identified the search is more focused to produce local optimal solutions in a process of 'intensification'. In many cases the differences between the various implementations of the Tabu method have to do with the size, variability, and adaptability of the Tabu memory to a particular problem domain. This article explores the meta-heuristic approach which is dramatically changing our ability to solve a host of problems in applied science, business and engineering. Tabu Search has important links to evolutionary and "genetic" methods, often overlooked, through its intimate connection with scatter search and path re-linking — evolutionary procedures that have recently attracted attention for their ability to facilitate the solution of complex problems. The adaptive memory designs of tabu search have also provided useful alternatives and supplements to the types of memory embodied in neural networks, allowing enhancements of neural network processes in practical settings.

Chapter 3

Research Methodology

3.0 Introduction

The method developed by Billington in 1986 was considered and description on the model used to compare the results. This chapter provides some descriptions on treatment of the interaction between production scheduling and lot-sizing. We will explore the fundamental model for both production scheduling and lot sizing based on Billington et. al and began with the introduction to Material Requirements Planning (MRP).

3.1 Material Requirements Planning

Material requirement Planning (MRP) is a computerized information system which determines the requirements for parts and components in multi-level multi-product production planning environment. The MRP system works under some logic such as: it takes a discrete production plan for a parent item from the master production system, explodes the parent items requirements into component items and raw materials, calculates the net requirements subtracting the available inventory from the gross requirements, then calculated the lot-sizes for all items at each stage, and finally offsets the lead times according to their due dates. Hence, the MRP system controls both the material control and planning at the same time.

There are three major paths in MRP system, which are: (a) Master Production Schedule (MPS), which is the input for the MRP systems, I.e. to determine the quantity and timing for each item to be produced, (b) Bill of Material (BOM) which provides the whole information about the components and end items within the hierarchical levels, which goes into that end product for the MRP, such as their sequence, their quantity in each finished items, (iii) Inventory Status File same called record file. It provides up-to-date information for each item. It involves an identification number; available quantity and procurement lead time of each item (Adam and Ebert [1989]). The outputs of the MRP system are the order release requirements, order rescheduling, and planned orders. Thus, production planner can meet the material requirements within the capacity and lead time context due to the output given.

3.2 The model components

Prescriptive versus descriptive models are the popular distinction concerns in modelling. The previous model focus on decision making whereas the descriptive models focus on evaluation of the performance of specified system (Dietrich, [1991]). While lot-sizing analysis may use a descriptive model, like simulation model, the focal point of lot sizing model is predominantly on decision making and action taking (R.Kuik et.al., [1994]).

Model parameters are quantities that are predetermined exogenously to the model and remain fixed during the model's execution: Their value is the output of the model. Therefore before we solve the problem, first we need to decide which quantities are parameter and which are decision variables. Parameter reflects conditions that are perceived stationary or predetermined during the decision routine.

The following elements are usually considered as parameters.

- ***Planning horizon and time scale***: it is the time interval on which assumptions are made with respect to demand and on which performance will be measured. Moreover, the planning horizon may be finite or infinite. Through R. Kuik et al [13]. an infinite planning horizon is usually accompanied by stationary demand and a finite planning horizon by dynamic demand. Then, time scale can be either continuous or discrete. ***Time Bucket*** is referring to some latitude for the exact timing of operations in the sense that planned production quantities are assigned to, e.g. days or weeks.

- ***Demand rate***: Demand is always consider as an exogenous quantity (input to the model). By R. Kuik et al. there are two broad lines of approach can be distinguished:
  - Models in which demand is considered stationary or even constant; (Where the demand is deterministic)
  - Models in which demand is considered dynamic.

(Time dependent demand is specified on a period-by-period base)

- ***Resource Constraints:*** A model is said to be uncapacitated model when capacity restrictions on resource are not restrictive, or modelled as cost components in the objective. On the other hand, a model is said capacitated, if the capacity constraints are explicitly stated.

- ***Lead-time*:** Generally lot-sizing model can have two type of lead-time. Once are exogenous (or nominal) lead times and next are endogenous lead times.

  Exogenous lead-times (ELT) can be due to the nature of a transformation process itself. The example exogenous lead-times given by R. Kuik el at was the drying of paint during some fixed time interval or the external factors such as purchasing lead-times. This kind of lead-time, are normally used in the MRP systems. ELT are referred to the outcome of the schedule of operations and thus of the batching decisions.

- ***The objective function:*** The generic objective is the minimization of the costs per unit of time.

The above discussion is large follow those given by R.Kuik et. al. [1994].

3.3 The Integer Formulation with bottlenecks problem

The work of Billington et al. [1986] provides a definitive treatment of types of lot-sizing problems. In this paper, the major part will be the suggestion by Billington et al. and some by Toklu and Wilson [1991] in this lot-sizing problem.

Through Billington et al. [1986], for the lot-sizing problem, a bottleneck well defined as follow; A bottleneck is a work centre that converts raw materials into finished goods. Under this definition, a machine with limited capacity, highly skilled or specialised workers, and task-specific machines or tools can all be seen to be bottlenecks. Where, all the resources could be classified into a bottleneck. A situation that when the work centre capacity is not enough to satisfy the demands for some periods is the problem that will discuss in this paper. A general production structure with a bottleneck facility is given in Figure 5(from Billington et al.[1986]).

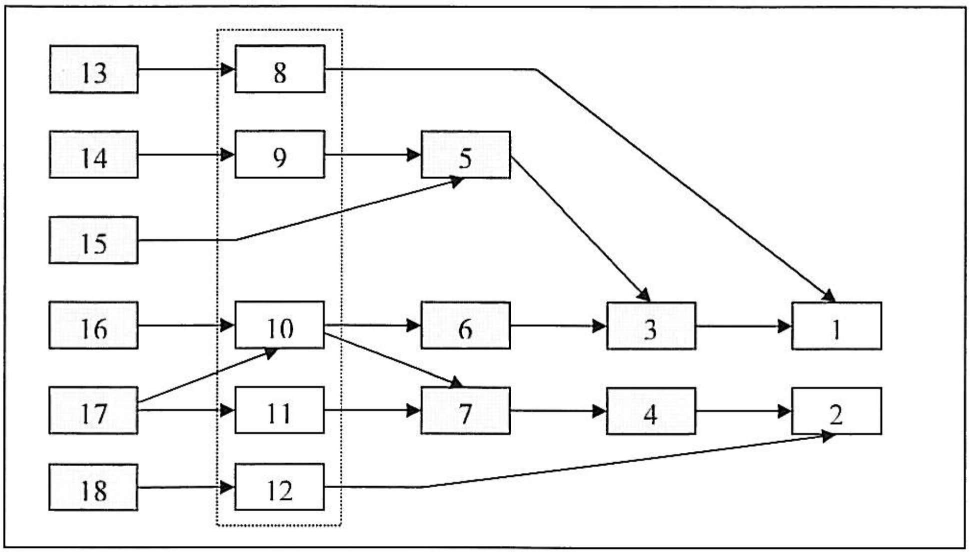


Figure 5: A general product structure with a bottleneck facility

From Figure 5 above, there is only one bottleneck facility occur in this product structure (dashed line). The numbers in Figure 5 illustrates the production items. Notes that, the numbering convention shown is that there is no item has a higher number than any of its predecessor.

The complex interdependencies, which exists between the planning decisions for different items was the causes make the multilevel lot-sizing problem (MLP) becomes difficult. The interdependencies in the production process are described by number, a_{ji}, where ***j*** and ***i*** range over the item numbers. a_{ji} meaning that the production one unit of ***j*** requires input of a_{ji} units item ***i***. Normally, the value of a_{ji} is described at the product structure of the planning problem. Where, item ***j*** defines a vertex ***j*** in the graph and the arc starting in ***j*** and ending in ***i*** whenever a_{ji}>0. As an example, look at the figure 1, the value of a_{ji} is show by the number in the grey colour, which is a_{ji}=1. From Billington el at.[3] a_{ji} is assume equal to zero, for all ***i*** ≥ ***j***. This assumption also use by many author too.

When considering two items ***i*** and ***j***, with a_{ji}>0, then ***j*** is the termed a **predecessor item** and the ***i*** is called a **successor item**. In an assembly product structure, each item will has precisely one successor item except one end item.

Several items are to be produced in order to meet some known dynamic demand without backlogs and stock outs. Furthermore, a production plan is aims at making possible the delivery of amount of items in periods in order to meet **independent demand** (external demand). Where, in our problem the independent demand for item ***j*** in period *t* will be denoted by d_{jt}, furthermore, when producing P_{it} of item *i* takes place in period *t* this triggers demand for $a_{ji}\ P_{it}$ items *j* in period $t-L_i$, where L_i is the lead time of the production of item *j*. Then this form of demand is called **dependent demand**. Let say, in Fig. 5 demands occur for item 8-12 is the dependent demand, which are the predecessor items for items 1-5. Normally, items 13-18 in Fig. 5 are purchased items which are use to make intermediate items, e.g. In Fig 5 item 13 is use to produce item 8,item 14 use to produce item 9, item 15 use to produce item 5, and so on. The item 1 and item 2 in Fig. 1 is the end item for this product. Therefore, to product item 1 and item 2 we required a_{31} units of item 3 to make item 1 and a_{42} units of item 4 to make item 2. However, we have to make sure that we have enough quantity of predecessor items for these successor items.

3.3.1 Assumption

Hence here is some assumption that we need to consider during our process for solving MLP with model. That is, we need to consider some assumption below:

1. All lead times between stages are assumed to be zero.
2. Demand for the multiple end items are assumed known and constants at any intermediate stages,
3. There is no demand for the component at any intermediate stages
4. Back order are not allowed,
5. The numbers of units coming from bill of material required in the production of one unit at the immediate successor stage to the other stage is assumed to be equal to one,
6. The unit production costs are assumed constant and hence are ignored

7. Production must occur in advance of that demand

3.3.2 The Mathematic model

The formulation below is the integer programming formulation for the lot-sizing problem with bottleneck. And it is the formulation when there are multiple capacity constrained work centres.

Where, the inventory here is eliminate by substituting cumulative production minus cumulative demand, and it is derived from a model in Billington et al. (1986).

Minimize:

$$Z = \sum_{j=1}^{J} \sum_{t=1}^{T} [h_j (T - t + 1) q_{jt} + cs_j X_{jt}] \quad (1)$$

Subject to:

$$\sum_{n=1}^{t} q_{j,n-v_j} - \sum_{i=1}^{t} a_{ji} q_{in}] \geq \sum_{n=1}^{t} d_{jn} - I_{j^\circ} \qquad j=1,2,\ldots,J \quad t=1,2,\ldots,T \quad (2)$$

$$\sum_{j=1}^{J} [b_{jl} q_{jt} + s_{jl} X_{jt}] \leq CAP_t \qquad l=1,2,\ldots,M \quad t=1,2,\ldots,T \quad (3)$$

$$X_{jt} = \begin{cases} 1; & \text{if } q_{jt} > 0 \\ 0; & \text{otherwise} \end{cases} \qquad j=1,2,\ldots,J \quad (4)$$

$$q_{jt} \geq 0 \qquad j=1,2,\ldots,J \quad t=1,2,\ldots,T \quad (5)$$

Where, the constraint (1) is the objective function for the problem, notes that the effect of the inventory cost is included as a production cost that decreases linearly with time. Thus, production that is shifted earlier than its respective demand will incur a holding cost based on the value added and proportional to the number of periods in stock. Normally, the objective function in an programming problem will includes the labour cost but in our case, the labour cost is constant, therefore we ignored it. The problem become a linear programming relaxation problem and is easily solved when they drop from the formulation. If not the integer programming may not give the feasible solution for some problems and takes substantial CPU time. The main ideal of this equation (objective function) is to find a trade-off between the holding and setup cost which minimise the total cost.

The second constraint is the constraint that makes sure that the cumulative production minus cumulative requirements in period ***t*** is always greater than or equal to the external demand in the planning horizon. It shows that, the production must be available at least n-v_j, where v_j is the lead time. Then $\sum_{i=1}^{t} a_{ji} q_{in}$ is the interrelation between the successor and predecessor items.

Constraint (3) is a capacity constraint for the bottleneck, and insures that setup cost and time are assessed only when there is production.

Thus, constraint (4) shows setup cost and time are applicable only if the is a production. Where, the value of X_{jt} is taken only value 0 and 1. Such a charge occurs when a production run to produce a batch of a particular product, q_{jt}, is undertaken and the required production facilities must be set up to indicate the run. It means that if there is no run of production for item i in t period, then the solution of X_{jt} is 0 and will take the value 1 if it the run is there. Where the value X_{jt} is use to calculate the setup cost for the particular run of the production.

Then finally is the constraint (5), it is a constraint that, make sure that q_{jt} is a non-negative decision variable.

Billington et al. [1986], present a heuristic method based on Lagrangian relaxation embedded within a branch and bound procedure, for the multi-level lot-sizing problem with a bottleneck. In their works, they propose the solution into two phases, which are dual and primal procedures. Furthermore, they define a subproblem by assigning a fixed value to some X_{jt} value at any node in the branch. Then relaxes all the capacity and all inventory balanced constraint for the production lot-size to solve it. Thus, a smoothing method is used to adjust the production in a primal phase. After that, they extend the primal phases to a dual phase, which yields modified setup costs and production in each period. Then repeat the primal phase with theses new price and iterate until a good solution is search. The procedure of this heuristic will not consider in this paper.

In this report, we will consider the heuristic that propose by Toklu and Wilson [1991] to solve the problem modelled by Billington et al. In their paper, they have concentrated on the three problem types: 1-end item, 3-end item and 5-end item.

3.3.3 A heuristic for multi-level lot-sizing problems with bottleneck(s)

Lot sizing in MRP only becomes realistic when features such as capacity constraints and the fact that systems are multi-level can be incorporated into the model. Blackburn and Millen [1982] review and add to contributions made to this area. Simultaneous lot sizing and capacity requirements planning in an MRP framework had been provides in their work. However one of the most successful attempts to tackle the multilevel lot-sizing problem with a bottleneck constraint has been done by Billington et al. [1986]. Therefore, this section will propose a simple heuristic with is develop by Bilal Toklu and J. M. Wilson. First, they dividing production items into end item and non-end items, e.g. in Fig 1 items 3-18 are the non-end items and item 1 and 2 are the end items. Then the items 3-7 and 13-18 are the non-end items outside bottleneck, thus the items 8-12 are the non-end item with bottleneck. The reason for this is that production of each non-end item is unconstrained and so has neither any effect on the production of any other non-end item nor the production of items which are constrained by the bottleneck.

As demand required of all items is known in advance, the production decision for each no-end –item becomes a relatively simple one of the when to produce in order to minimize the contribution to costs (holding and setup costs) of each non-end-item. The fact that demand for end-items determined the demand for intermediate items does not invalidate the independence of the production of each item as demand for end-items is known several periods in advance. The extension to dependent product item cases for different product structures is an area for further research. To determine when to produce end- items is more complex as these product items must share the resources of the bottleneck. Thus, for these items the production problem is a constrained problem. However, in general these items are in the minority.

Define S_{jt} as stock of product j at start of period t, and P_{jn} as the quantities of product i at the end of period t

then $$S_{jt} = \sum_{t=n}^{t-1} P_{jn} - d_{jn} \qquad t = 2,3,\cdots$$

where $S_{j1} = S_{j0}$

3.3.3.1 Non-end items outside bottleneck

For these product items the EOQ and the Silver-Meal approaches will be used. These approaches were chosen as they are comparatively simple to operate and in general will produce solutions of good quality.

3.3.3.1.1 The Economic Order quantity Approach (EOQ)

Let Q_j be the EOQ for product item j, based on the setup cost cs_i and the holding cost h_j. Then the following strategies are considered:

(a) Produce Q_j in period 1 and then next product Q_j in the period when stocks would become negative if no production were made (ie. Find the next smallest t such that $S_{jt} < d_{jn}$)

Let t_j be the number of occasions on which product item j will be produced. Then

$$t_j = [\sum_{t=1}^{T} d_{jt} / Q_j + 0.5]$$

and the production is made in any period n whenever $S_{jt} < d_{jn}$. Note that [] is the integer part function.

If in any period *n*

$$Q_i \geq \sum_{t=n}^{T} d_{it} - S_{in}$$

then set

$$Q_j = \sum_{t=n}^{T} d_{jt} - S_{jn}$$

(b) Let Z_j be the quantity of product item I produced in period 1. The same quantity is next produced whenever stocks would become negative if no production were made (ie. When $S_{jt} < d_{jn}$).

$$Z_j = \left[\sum_{t=1}^{T} d_{jt}\right] \Big/ t_j$$

Continue this process through all the periods.

(c) Produce all product items in period 1. Strategies (a), (b), and (c) are evaluated to see which leads to the smallest total inventory cost over the N periods and then that strategy is chosen.

3.3.3.1.2 The silver-meal Approach (SM)

This approach selects the lot-size that covers the number of periods to minimise the total cost per period. The average total cost function of this heuristic given as:

$$\text{Average total Cost} = \left\{ cs_j + h_j \sum_{t=1}^{T} (t-1) d_{jt} / t \right\}$$

And chooses the smallest period t whenever the average total cost is greater than the immediate previous periods cost (i.e. The average total cost of (t) > the average total cost of (t-1)). Then the lot-size equals

$$Q_j = \sum_{t=n}^{N} d_{jt}$$

Where j is the product items number.

3.3.3.2 Non- End items with Bottleneck

Assume that there is a bottleneck located in the intermediate level. I.e. bottleneck facility located in 3rd item in 1-end-item problem. To deal with this kind of problem, note that the non-bottleneck items have been explained for different kinds of end-item problems. Therefore in this section the bottleneck items in the non-end-items will be explained.

For this product items a simple heuristic was adopted which would adopt a greedy approach to production by having few setups, but with heavy utilisation of the resultant production capacity. In addition, the heuristic would operate in a cyclic manner, moving between produce reasonably smooth productions. The approach has broad similarities with the work of McClain and Trigerio [1985] except that by excluding setup time and cost they handle a problem that is easier to solve. Bahl and Ritzman [1984] also adopt a cyclic approach but do so by examining permutation schedules.

The heuristic will describe with reference to three cases. However, in this project we only discusses about two cases only, with is 1-end-item problem and the 3-end-item problem.

Case (a) 1-end-item problems

Assume that there is one bottleneck has been located in the 3rd item in the product structure, the heuristic is that which produces as much of non-end-item j as cap_t will allow in period 1 i.e. set $P_{j1} = cap_1$, then next produce j when stocks would become negative if no production were made, i.e. find the next smallest t for which $S_{jt} < d_{jn}$.

Continue the process of producing in each period t, which has this property.

If P_{jn} would exceed $\sum_{t=n}^{T} d_{jt}$ for any period n

Then set $P_{jn} = \sum_{t=n}^{T} d_{jt}$

Case (b) 3-end-item problems

Assume that the bottleneck is located at the 7th, say items in the product structure and according to the heuristic, the first priority is in the first item in the structure, then the second item and so on.

A three period cycle is adopted.

Period 1: Set $P_{21} = d_{21}$, $P_{31} = d_{31}$ and $P_{11} = cap_1 - d_{21} - d_{31}$

Period 2: Set $P_{32} = d_{32}$, $P_{12} = 0$ and $P_{21} = cap_2 - d_{32}$ provided $S_{12} > d_{12}$

Otherwise set $P_{12} = d_{12} + d_{13} - S_{12}$, $P_{32} = d_{32}$ and $P_{22} = cap_2 - P_{12} - P_{32}$

Period 3: Set $P_{33} = cap_3$ and $P_{13} = P_{23} = 0$ provided $S_{13} > d_{13}$ and $S_{23} > d_{23}$

Otherwise set $P_{13} = d_{13} - S_{13}$, $P_{23} = d_{23} + d_{24} - S_{23}$ and $P_{33} = cap_3 - P_{13} - P_{23}$

Period 4: Set $P_{14} = cap_4$ provided $S_{24} > d_{24}$ and $S_{34} > d_{34}$

Period 5: Set $P_{15} = cap_4$ provided $S_{15} > d_{15}$ and $S_{35} > d_{35}$

Period 6: Set $P_{36} = cap_6$ provided $S_{16} > d_{16}$ and $S_{26} > d_{26}$

Again, if stocks of any product would become negative, produce sufficient of that product to satisfy demand over the next one or more periods until that product moves into the dominant production position. The product in the dominant production position is the product for which as much as possible should be produced after ensuring stocks of other products do not become negative. The heuristic above shown that product 1 is in the dominant production position in periods 1, product 2 is in the dominant production position in period 2 and so that product 3 is in the dominant production position at period 3. Whenever the stocks of any product would become negative, produce as much of that product item to satisfy the demands according to their priority. Continue the process in the same cyclic manner for the remaining periods. If at any stage stocks of all products are sufficient for production to be zero in any period, no production is made in that period and the cycle for the appropriate product is delayed by one period.

3.3.3.3 End Item with bottleneck

If there is a bottleneck facility occurs at the end item, the heuristic rules that use to solve this problem is the same as we have discuss in the section of non-end-item with bottleneck. The only different is : instead of item(s) 7^{th}, 1^{st} items will be used in the three end-item problem, in the one end-item problem item 1 will be the bottleneck instead of item 3, which is grouped in the same way as previously.

There are some cases studies have been done to show how the system work in the next chapter.

3.3.4 The Mathematic Model Develop By Alf Kimms

The model below is presented by Al Kimms. As we know that several items are to be produced in order to meet estimated dynamic demand without backlogs and stock outs. Generally the relations among those items defined an acyclic goes into structure as we have show at previous chapter.

As we have seen that in the Billington et al model, the demand only allow for end items only. Furthermore, this assumption also use by many authors. However, in the model above, the demand occur for all items including component parts are allowed. Where, the model allows us to have a production schedule that meet the external demand occur.

As same as other model, Al Kimms state that there is some basic assumptions need to assume for this model. First, is about the demand as we have already mentioned just now, the demand may occur for all items including component parts. Then, the finite planning horizon is subdivided into a number of discrete time periods. The lead times should be a positive number, which are given due to technological restrictions such as cooling or transportation for instance. Furthermore, items share common resources. Therefore, may be all (some) of them are scarce. The capacities may vary over time. And to produce an item requires an item-specifies amount of the available capacity. And most of all, all data are assumed to be deterministic.

As we know that items, which are produced in a period some time, is use to met some future demand. Therefore, it must be stored in inventory and thus it cause item-specific holding costs. Each item requires at least one resource for which are assumes to be sequence independent will incur because of setting a resource up for producing a particular item. However setup time are not considered. Hence, some items, which are produced having some idle time in-between, do not enforce more than one setup action.

Through, Al Kimms the most fundamental assumptions here is that for each resource at most one setup may occur within one period. Hence, at most two items sharing a common resource for which a setup state exists may be produced per period. A problem is known as the proportional lot sizing and scheduling problem, due to this assumption.

3.3.4.1 Multi-level lot sizing and scheduling

Models:

$$\text{Min} \sum_{j=1}^{J} \sum_{t=1}^{T} (s_j x_{jt} + h_j I_{jt}) \qquad (45)$$

Subject to

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} - \sum_{i \in S_j} a_{ji} q_{it}, \qquad j = 1,\ldots,J, \quad t = 1,\ldots,T, \tag{46}$$

$$I_{jt} \geq \sum_{j \in S_j} \sum_{\tau = t+1}^{\min\{t+v_j, T\}} a_{ji} q_{i\tau}, \quad j = 1,\ldots,J, \quad t = 0,\ldots,T-1, \tag{47}$$

$$p_j q_{jr} \leq C_{m_j t}(y_{j(t-1)} + y_{jt}), \quad j = 1,\ldots,J, \quad t = 1,\ldots,T, \tag{48}$$

$$\sum_{j \in \Gamma_m} p_j q_{jt} \leq C_{mt}, \quad t = 1,\ldots,T, \quad m = 1,\ldots,M, \tag{49}$$

$$\sum_{j \in \Gamma_m} y_{jt} \leq 1, \quad m = 1,\ldots,M, \quad t = 1,\ldots,T, \tag{50}$$

$$x_{jt} \geq y_{jt} - y_{j(t-1)}, \quad j = 1,\ldots,J, \quad t = 1,\ldots,T, \tag{51}$$

$$y_{jt} \in \{0,1\}, \quad j = 1,\ldots,J, \quad t = 1,\ldots,T, \tag{52}$$

$$I_{jt}, q_{jt}, x_{jt} \geq 0, \quad j = 1,\ldots,J, \quad t = 1,\ldots,T, \tag{53}$$

As we have mention early, in our case we are consent about the capacitated lot-sizing problem (CLSP). It was a problem where many items can be produced per period. Therefore, we need to derive the model that shown before, so that, the capacitated condition is taking account. There is a most important thing that we need to remember that CLSP does not include sequence decisions.

Before, we discuss about the transformed the model, we like to discuss about the constraints that show at the model. As we can see, equation (45) is the objective function, where the main ideal is to minimize the sum of setup and holding costs. Equation (46) is the inventory balances. Where, the total inventory in hand at the end of period *t* was the number of items remanded at the end of period *t*-1 plus the items produced minus the external and internal demand. As a remainder, to meet the internal demand we must respect a positive lead times. Normally, people will assume the lead-time as zero so that we do in ours project. And constraint (47) is only use to make sure that the inventory in the period *t* satisfies this condition. Then, the constraint (48) was the constraints, which make sure that there is a proper setup state either at the beginning or at the end of a particular period, so that production can take place. The constraint (49) is the capacity constraints. Since the right hand side is a constant, overtime is not available. (50) is the constraint which make sure that the setup state of each machine is uniquely defined at the end of each period. The constraint (51) shows us which setup happens in each period. (52) defined the binary-valued setup state variables, while (53) are simple non-negativity condition.

Therefore, to do transform, first we need to drop (50). This is because in this case, we no need to make sure that the setup state of each machine is uniquely defined at the end of each period. Then, the following constraints

$$p_j q_{jt} \le C_{m_j t} y_{jt}, \quad t = 1,\ldots,T, \quad j = 1,\ldots,J \tag{54}$$

and

$$x_{jt} = y_{jt}, \qquad j = 1,\ldots,J, \quad t = 1,\ldots,T, \tag{55}$$

are introduced instead. Note that the setup costs are charged now in every period in which production takes place. Again, the resulting model can be drastically reduced. (51) and (48) are now becomes redundant. Additionally, we can eliminate the variables x_{jt} by substitution. Usually, $v_j = 0$ is assume for all or some j = 1,...j since we have large (time) buckets in mind. This makes (47) obsolete for these items. It should be clear that once we give up (50) the resulting model does not support sequence decisions any more.

Therefore, the model for ours problem is:

$$\text{Min} \sum_{j=1}^{J} \sum_{t=1}^{T} (s_j x_{jt} + h_j I_{jt})$$

Subject to

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} - \sum_{i \in S_j} a_{ji} q_{it},$$

$$j = 1,...,J, \quad t = 1,...,T, \tag{46}$$

$$p_j q_{jt} \le C_{m_j t} y_{jt}, \quad t = 1,...,T, \quad j = 1,...,J \tag{54}$$

$$\sum_{j \in \Gamma_m} p_j q_{jt} \le C_{mt}, \quad t = 1,...,T, \quad m = 1,...,M, \tag{49}$$

$$x_{jt} = y_{jt}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{55}$$

$$y_{jt} \in \{0,1\}, \quad j = 1,...,J, \quad t = 1,...,T, \tag{52}$$

$$I_{jt}, q_{jt}, x_{jt} \ge 0, \quad j = 1,...,J, \quad t = 1,...,T, \tag{53}$$

Note that, the equation (53) and (54) can combined as one equation. Therefore, make it simple we can just substitute $x_{jt} = y_{jt}$ into (54).

This is the integer programming formulation for the CLSP problem, which we will used to compare the result given with the result given by Billington el at model and by the heuristic method.

Chapter 4

Modelling and Case Study

4.0 Introduction

In this report, we reproduced the result as proposed by Billington el at., (1983) and compare it against the model developed by Al Kimms (1996). He we present a single level problem as no item is predecessor or successor of another item and some multilevel problems, as a test case. The value of the initial parameters as developed similarly to the set data used by Billington el at., 1986. We consider a family of *N* items, which are produced in or obtained from the same production facility. The external demands are assumed known for all the end products and must be satisfied without backorders over a finite horizon of *T* periods with no external demand for components. Here we exclude the sequencing of production within the time period and the possibility of carrying over a setup between periods. The aggregate order size is constrained by a capacity limit. The objective is to find a lot-sizing strategy that gives or satisfies the demand for all items over the entire horizon without backlogging, and which minimizes the total cost of production run. This is the sum of the total holding and setup costs. All demands, cost parameters and capacity limits may be time-dependent.

In this chapter, we will recall the test problem and the result given by using heuristic model follow by the result given by Al Kimms. Finally, some discussions on both models are presented.

4.1 Experimental Study

4.1.1 The Single level multi item capacitated lot size problem

A simple example is used to explore the knowledge of how the system works. Suppose we have a single level multi item capacitated lot size problem with a bottleneck facility. Assume that all of the items are independent, and all items are produced under one bottleneck facility.

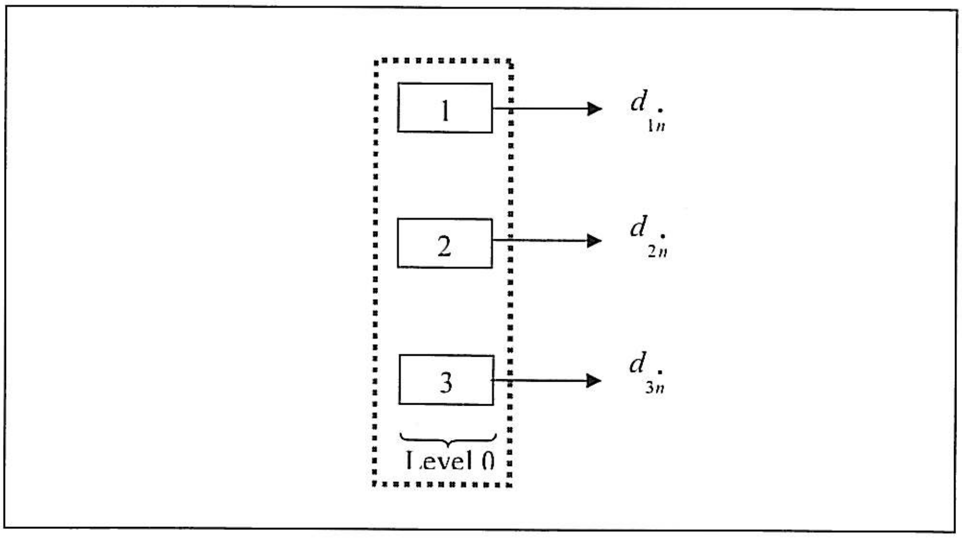


Figure 6: single level multi items product structure

Supposes that, J is the total size of the items and T is the total number of periods in the problem and a_{ji} is the quantity of item j that is directly needed to produce one item i, i = 1, 2, 3 ...n and j = 1, 2 ... m. If J = 3, T = 6, and all a_{ji} = 0, for $\forall j$. The maximum number of each items, $N_{\max j}$ can be produce in each period t are 60 for $\forall j$. Assume that the initial inventory for all item are assume to be zero and similarly for lead-time. If d_{jt} is the number of items demanded by period t, and assume that the numbers of external demand only exists for end items. Since we have 3 end item (Fig. 6), so the external demand occur for each item are:

For item1, there is no external demand at period 1, then there are 10 units of item 1 need to meet the demand at period 2, 3 and 4, or we can write as d_{12} is 10, d_{13} is 10, d_{14} is 10. Thus external demand for item 1 at period 6 is 20 units, d_{16} is 20.

For item 2, the demands occur at period 1 is 10 units and the number is same for period 2 and 5. There is no demand occur for periods 3 and 4. It can write as d_{21} is 10, d_{22} is 10 and d_{25} is 10. The external demands occur for item 3 are 10 units at period 1, 3 and 6 each. There is no external demand occur at period 2, 4 and 5. Thus we can write the parameter as d_{31} is 10, d_{33} is 10 and d_{36} is 10.

Furthermore, other parameters include the production time of *j* unit item on the bottleneck facility, b_j. For example to produce one unit of item 2 is 1 minutes then b_2 = 1 min. Here we assume b_j = 1 minutes, for $\forall j$ and the setup time, S_j = 1 min for $\forall j$, for the work centre for product *j* in the bottleneck facility.

Suppose that, we set the holding cost for item *j* = 1, 2, 3, as 0.5, 2, 1. Then we set the setup cost for item *j* = 1, 2, 3, as 50, 100, 190 and the capacity utility is approximate 90 percents. Assume that CAP_t is the available capacity of the work centre at time *t*, for *t* = 1, 2 ...T, let say here we set that capacity available for period 1, CAP_1 is 60, CAP_2 is 60, CAP_3 is 60, and CAP_4 is 60.

We found that, the limitation of the capacity exists in this problem and given v_j = 0, as v_j is a positive and integral lead-time of item *j* or calls safety lead-time for product *j*, for $\forall j$. It satisfies the condition for a bottleneck problem that exited in a production line. This gives us a reason to suggest that this v_j is a lot-sizing problem with bottleneck.

In this problem, a_{ji}, the quantity of item *j* needed to produce one item of product *i* is equal to zero (a_{ji} = 0). This mean that there is no relationship between item *j* in the production of item *i* or in other words, it means that there is no units of item *j* required directly in the production of one unit of item *i* since we are consider a single level multi item capacitated lot size problem.

Furthermore, in this problem I_{jt} is the inventory for item *j* at the end of period *t*. Thus, I_{jo} is the initial inventory for item *j*, assume that in this case I_{jo} is equal to zero (I_{jo} = 0), and meaning that there is on inventory in hand remaining from the previous periods before period 1.

We consider the problem to have 6 periods. In each period, it will produce 3 items with number of product, q_{jt} for *j* = 1, 2, 3, and *t* = 1, 2, 3, 4, 5, 6 to minimise total cost. Time needed on the bottleneck facility for the production of product ***j***, b_j and setup time for the work centre for product ***j***, S_j are the variable, which will exist in a bottleneck facility. The most important constraint considers here is the capacity of the production in each period. Here the total capacity in the first period is less than or equal to 60. Then, the capacity for the second, third and fourth period until period 6 is also less than or equal to 60.

To solve this problem, we consider the model derived from model (1), with heuristic method and from model (2). These models are the Integer Programming formulation as described in Chapter 3. We proposed the solution to the above models using LINDO software.

The following section describes the implementations of LINDO in solving the problem.

4.1.1.1 The implementation in LINDO for model (1):

The solution to the above models may be solved using on the shelves software known as LINDO. It began with the opening of a new blank LINDO Model Window and saves it as a file. Then we developed the objective of the model and define the variables of the problem as given by Billington et al (1986), which is an IP formulation. The model may be described as follows:

The objective function is:

(a) Minimize Z =

$$3q_{11} + 2.5q_{12} + 2q_{13} + 1.5q_{14} + 1q_{15} + 0.5q_{16} +$$
$$50X_{11} + 50X_{12} + 50X_{13} + 50X_{14} + 50X_{15} + 50X_{16} +$$
$$12q_{21} + 10q_{22} + 8q_{23} + 6q_{24} + 4q_{25} + 2q_{26} +$$
$$100X_{21} + 100X_{22} + 100X_{23} + 100X_{24} + 100X_{25} + 100X_{26} +$$
$$6q_{31} + 5q_{32} + 4q_{33} + 3q_{34} + 2q_{35} + q_{36} +$$
$$90X_{31} + 90X_{32} + 90X_{33} + 90X_{34} + 90X_{35} + 90X_{36}$$

Where x_{jt} is the binary variable, which indicates whether a setup for item *j* occurs in period *t* for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T. It is taken the value 0 or 1, $(x_{jt} = 1)$ or not $(x_{jt} = 0)$, and q_{jt} is the production quantity for item *j* in period *t* for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T. I_{jt} is the inventory for item *j* at the end of period *t*, for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T.

The variable and the constraints:

From the previous section, we have assume that a_{ji} = 0, v_j = 0, and I_{jo} = 0, and given that $d_{12} = d_{13} = d_{14} = 10, d_{16} = 20$, $d_{21} = d_{22} = d_{25} = 10$, $d_{31} = d_{33} = d_{36} = 10$. Since J=3, T=6, the equation

(b) $$\sum_{n=1}^{t} q_{j,n-v_j} - \sum_{i=1}^{t} a_{ji} q_{in}] \geq \sum_{n=1}^{t} d_{jn} - I_{j^o} \qquad \begin{matrix} j=1,2,\ldots,J \\ t=1,2,\ldots,T \end{matrix}$$

Becomes,

$$\sum_{n=1}^{6} q_{jn} \geq \sum_{n=1}^{6} d_{jn} \qquad \text{for} \quad \begin{matrix} j=1,2,3, \\ t=1,2,\ldots,6 \end{matrix} \qquad (1)$$

From the information given we knew that

$$\sum_{n=1}^{6} d_{1n} = d_{1n}^{\cdot} = 50$$

$$\sum_{n=1}^{6} d_{2n} = d_{2n}^{\cdot} = 30$$

$$\sum_{n=1}^{6} d_{3n} = d_{3n}^{\cdot} = 30$$

So expending (1), we have

$$q_{11} + q_{12} + q_{13} + q_{14} + q_{15} + q_{16} \geq d_{1n}^{\bullet} = 50$$

$$q_{21} + q_{22} + q_{23} + q_{24} + q_{25} + q_{26} \geq d_{2n}^{\bullet} = 30$$

$$q_{31} + q_{32} + q_{33} + q_{34} + q_{35} + q_{36} \geq d_{3n}^{\bullet} = 30$$

(c) from $\sum_{j=1}^{N} [b_j P_{jt} + s_j X_{jt}] \leq CAP_t$ $\qquad t = 1,2,\ldots,6$

we have;

$$q_{11} + X_{11} + q_{21} + X_{21} + q_{31} + X_{31} \geq 60$$
$$q_{12} + X_{12} + q_{22} + X_{22} + q_{32} + X_{32} \geq 60$$
$$q_{13} + X_{13} + q_{23} + X_{23} + q_{33} + X_{33} \geq 60$$
$$q_{14} + X_{14} + q_{24} + X_{24} + q_{34} + X_{34} \geq 60$$
$$q_{15} + X_{15} + q_{25} + X_{25} + q_{35} + X_{35} \geq 60$$
$$q_{16} + X_{16} + q_{26} + X_{26} + q_{36} + X_{36} \geq 60$$

(d) $X_{jt} = 1$; if $\begin{cases} q_{jt} > 0 \\ 0 ; \text{ otherwise} \end{cases}$ $\qquad j = 1,2,\ldots,J$

(2)

From the Chapter 3, it is shown that X_{jt} is the production indicator, where it is a kind of the fixed-change problem. In this case the problem is to reformulate the structure of the model so that it can be solved using IP. The constraints is written as :

$q_{11} - 60X_{11} \leq 0$ $\qquad q_{16} - 60X_{16} \leq 0$
$q_{12} - 60X_{12} \leq 0$ $\qquad q_{21} - 60X_{21} \leq 0$
$q_{13} - 60X_{13} \leq 0$ $\qquad q_{22} - 60X_{22} \leq 0$
$q_{14} - 60X_{14} \leq 0$ $\qquad q_{23} - 60X_{23} \leq 0$
$q_{15} - 60X_{15} \leq 0$ $\qquad q_{24} - 60X_{24} \leq 0$

q_{25} - $60X_{25}$ ≤ 0

q_{26} - $60X_{26}$ ≤ 0

q_{31} - $60X_{31}$ ≤ 0

q_{32} - $60X_{32}$ ≤ 0

q_{33} - $60X_{33}$ ≤ 0

q_{34} - $60X_{34}$ ≤ 0

q_{35} - $60X_{35}$ ≤ 0

q_{36} - $60X_{36}$ ≤ 0

After the transformation, then only it can be implemented using LINDO. LINDO will generate the iterations and provide the result as given in the following section.

4.1.1.2 The Result by Billington model

Then the result given from LINDO package is:

item \ period	1	2	3	4	5	6
1	-	-	-	-	50	-
2	-	8	11	11	-	-
3	-	-	-	-	-	30

Table 2: Quantity produced in each period

Table 2 shows the quantity produced for each items in each period. It gave the quantity $q_{15} = 50$, $q_{22} = 8$, $q_{23} = 11$, $q_{24} = 10$, and $q_{36} = 30$. The empty space means that there is no production run in that period for certain item.

item \ period	1	2	3	4	5	6
1	-	-	-	-	1	-
2	-	1	1	1	-	-
3	-	-	-	-	-	1

Table 3: Binary Variable indicator for Setup

Table 3 shows the binary variable indicator for setup in each period. Here X_{15}, X_{22}, X_{23}, X_{24}, X_{36} is taken the value of 1 and the rest is 0 (the X_{jt} which taken the value 0 is show as the blank column). Thus the total cost is:

Total cost = 1(50)+10(8)+8(11)+6(11)+1(30)+100(3)+90(1)

= RM754.00.

The feasible solution is as follows - in period 2 the production run produces 8 units of item 2, and 11 units of item 2 in period 3 and period 4, 50 unit of item 1 at period 5 and 30 units item 3 in period 6 with the total cost RM745. In this finding, we found that not all products are produced in the same periods. We could not produce all item in the same period because there are the capacity limited in each period, therefore it is impossible to product all item at a same period as it will exceed the capacity limit.

4.1.1.3 The implementations with our proposed heuristic

From the previous section, given that $d_{12} = d_{13} = d_{14} = 10, d_{16} = 20$, $d_{21} = d_{22} = d_{25} = 10,$ $d_{31} = d_{33} = d_{36} = 10,$ and assume that all $S_{j1} = 0$ for $\forall j$. Where S_{jt} is stand for the stock of product *j* at the beginning of period *t*, and P_{jn} as the quantities of product *i* at the end of period *t*.

Assume that $cap_t = 60,$ and assume that 1-minute is needed to produce each unit of item *j*. Therefore, the implementation of the solution for this question is shown as below:

Step 1: *Period 1: Set* $P_{21} = d_{21},$ $P_{31} = d_{31}$ and $P_{11} = cap_1 - d_{21} - d_{31}$,

Thus we have,

$$P_{21} = 10, P_{31} = 10$$

$$P_{11} = cap_1 - d_{21} - d_{31} = 60 - 10 - 10 = 40, \text{ since } P_{11} < \sum_{t=1}^{6} d_{1t} = 50$$

It meaning that produced 40 units of item 1 will not cause the waste of recourse since the total external demand still need to fulfil in the following periods is more than the numbers we produce now.

Therefore, the stock in hand for the beginning of the next periods, for all items is given as:

$$S_{jt} = \sum_{n=0}^{t-1} P_{jn} - d_{jn} \qquad t = 2,3,\cdots$$

Thus,

$$S_{12} = \sum_{n=0}^{1} P_{1n} - d_{1n} \qquad S_{22} = \sum_{n=0}^{1} P_{2n} - d_{2n} \qquad S_{32} = \sum_{n=0}^{1} P_{3n} - d_{3n}$$

$$= S_{10} + P_{11} - d_{11} \qquad = S_{20} + P_{21} - d_{21} \qquad = S_{30} + P_{31} - d_{31}$$

$$= 0 + 40 - 0 = 40 \qquad = 0 + 10 - 10 = 0 \qquad = 0 + 10 - 10 = 0$$

Thus, in this period the production run produces 40 units of item 1, and 10 units each for item 2 and 3. Furthermore, there still have 40 units of item 1 as the inventory in hand for the next periods. Thus the is no item 2 and 3 left in the end of period 1 because all the units item produced in this period have been use to fulfil the external demand that occur in that particular period.

Step 2: *Periods 2: Set* $P_{32} = d_{32,}\ P_{12} = 0$ and $P_{22} = cap_2 - d_{32}$ provided $S_{12} > d_{12}$, *otherwise set* $P_{12} = d_{12} + d_{13} - S_{12}$, $P_{32} = d_{32}$ and $P_{22} = cap_2 - P_{12} - P_{32}$

In this period, we have 40 units of item 1 as the stock of item 1 in period 1. As $S_{12} = 40$. Since $S_{12} = 40 > d_{12} = 10$,

Thus we set $P_{32} = 0,\ P_{12} = 0$ and

$$P_{22} = cap_2 - d_{32}$$
$$= 60 - 0 = 60,$$

Since $P_{22} > \sum_{t=2}^{6} d_{2t} = 20$, therefore, we set $P_{22} = 20$

We do this because there is wastage occurs if we produced 60 units of item 2 because the total demand for the next period is only 20 units. This will automatically increase the total production cost.

Therefore, the stock in hand for the beginning of the next periods, for all items is given as:

$$S_{13} = \sum_{n=0}^{2} P_{1n} - d_{1n} \qquad S_{23} = \sum_{n=0}^{2} P_{2n} - d_{2n} \qquad S_{33} = \sum_{n=0}^{2} P_{3n} - d_{3n}$$

$$= S_{12} + P_{12} - d_{12} \qquad = S_{22} + P_{22} - d_{22} \qquad = S_{32} + P_{32} - d_{32}$$

$$= 40 + 0 - 10 \qquad = 0 + 20 - 10 \qquad = 0 + 0 - 0$$

$$= 30 \qquad = 10 \qquad = 0$$

In this period the production run produces 20 units of item 2, and there is no production of item 1 and 3, since the stock of item 1 still sufficient to fulfil the demand of item 1 and there is no external demand occurs for item 3. Thus, the stock in hand of item 1 becomes 30 units and item 2 is at 10 units.

Step 3: Periods 3: *Set* $P_{33} = cap_3$ *and* $P_{13} = P_{23} = 0$ *provided* $S_{13} > d_{13}$ *and* $S_{23} > d_{23}$, otherwise set $P_{13} = d_{13} - S_{13}, P_{23} = d_{23} + d_{24} - S_{23}$ *and* $P_{33} = cap_3 - P_{13} - P_{23}$

From the previous step we set $S_{13} = 30$, $S_{23} = 10$, $S_{33} = 0$, Therefore it satisfied the condition $S_{13} = 30 > d_{13} = 10$ and $S_{23} = 10 > d_{23} = 0$. Therefore, we have

$$P_{33} = 60 \text{ and } P_{13} = P_{23} = 0$$

Since $P_{33} > \sum_{t=3}^{6} d_{3t} = 20$, thus we set $P_{33} = 20$

Since we have satisfied the condition of the model, thus the production run in this period only produces 20 units of item 3, and no production run for item 1 and two.

Therefore, the stock in hand for the beginning of the next periods, for all items is given as:

$$S_{14} = \sum_{n=1}^{3} P_{1n} - d_{1n} \qquad S_{24} = \sum_{n=1}^{3} P_{2n} - d_{2n} \qquad S_{34} = \sum_{n=1}^{3} P_{3n} - d_{3n}$$

$$= S_{13} + P_{13} - d_{13} \qquad = S_{23} + P_{23} - d_{23} \qquad = S_{33} + P_{33} - d_{33}$$

$$= 30 + 0 - 10 \qquad = 10 + 0 - 0 \qquad = 0 + 20 - 10$$

$$= 20 \qquad = 10 \qquad = 10$$

The schedule in this period is only need to produce 20 units of item 3.

Step 4: *Periods 4: Set* $P_{14} = cap_4$ provided $S_{24} > d_{24}$ and $S_{34} > d_{34}$

Here we have $S_{14} = 20$, $S_{24} = 10$, $S_{34} = 10$, since $S_{24} = 10 > d_{24} = 0$ and $S_{34} = 10 > d_{34} = 0$,

Thus we have

$$P_{14} = 60$$

But $60 > \sum_{t=4}^{6} d_{1t} - S_{14} = 30 - 20 = 10$, so we set $P_{14} = 10$.

Therefore, the stock in hand for the beginning of the next periods, for all items is given as:

$$\begin{aligned} S_{15} &= \sum_{n=1}^{4} P_{1n} - d_{1n} \\ &= S_{14} + P_{14} - d_{14} \\ &= 20 + 10 - 10 \\ &= 20 \end{aligned} \qquad \begin{aligned} S_{25} &= \sum_{n=1}^{4} P_{2n} - d_{2n} \\ &= S_{24} + P_{24} - d_{24} \\ &= 10 + 0 - 0 \\ &= 10 \end{aligned} \qquad \begin{aligned} S_{35} &= \sum_{n=1}^{4} P_{3n} - d_{3n} \\ &= S_{34} + P_{34} - d_{34} \\ &= 10 + 0 - 0 \\ &= 10 \end{aligned}$$

Then in this period, we went back to check the quantity of item 1 again to see whether it still enough to satisfy the demand for the next few period. Since the total demand for item 1 in the last two periods is more than the stock, so we need to continue to produce the number of item 1.

Step 5: *Period 5: Set* $P_{25} = cap_5$ provided $S_{15} > d_{15}$ and $S_{35} > d_{35}$

At the beginning of the period 5, we have $S_{15} = 20$, $S_{25} = 10$, . $S_{35} = 10$.

Since $S_{15} = 20 > d_{15} = 20$ and $S_{35} = 10 > d_{35} = 10$

Therefore we set $P_{25} = 60$,

but $P_{25} = 60 > \sum_{t=5}^{6} d_{2t} - S_{24} = 10 - 10 = 0$, thus set $P_{25} = 0$

Therefore, the stock in hand for the beginning of the next periods, for all items is given as:

$S_{16} = \sum_{n=1}^{5} P_{1n} - d_{1n}$ $\quad S_{26} = \sum_{n=1}^{5} P_{2n} - d_{2n}$ $\quad S_{36} = \sum_{n=1}^{5} P_{3n} - d_{3n}$

$= S_{15} + P_{54} - d_{15}$ $\quad = S_{25} + P_{25} - d_{25}$ $\quad = S_{35} + P_{35} - d_{35}$

$= 20 + 0 - 20$ $\quad = 10 + 0 - 10$ $\quad = 10 + 0 - 0$

$= 0$ $\quad = 0$ $\quad = 10$

Notes that since all stock in hand still enough for us to fulfil the demand. Thus, there is no production run needed.

Step 6: *Period 6: Set* $P_{36} = cap_6$ provided $S_{16} > d_{16}$ and $S_{26} > d_{26}$

Since $S_{16} = 20 > d_{16} = 20$ and $S_{26} = 0 > d_{26} = 0$

So set $P_{36} = 60$, however $60 > \sum_{t=5}^{6} d_{3t} - S_{36} = 10 - 10 = 0$

So we set $P_{36} = 0$

At the end of periods of our production plan, usually there are nothing left. This is because of the objective function because the main ideal is to minimise the cost. Thus, the planning is done under the condition that there is no waste of resource during the time horizons T.

Generally, the result that we obtain from the calculation can be summaries as shown by the table below:

item \ period	1	2	3	4	5	6
1	40	-	-	10	-	-
2	10	20	-	-	-	-
3	10	-	20	-	-	-

Table 4: Quantity produced in each period

item \ period	1	2	3	4	5	6
1	-	40	30	20	20	-
2	-	-	10	10	10	20
3	-	-	-	10	10	-

Table 5: Inventory in hand at the beginning of period

period / item	1	2	3	4	5	6
1	1	-	-	1	-	-
2	1	1	-	-	-	-
3	1	-	1	-	-	-

Table 6: Binary Variable indicator for Setup

With the total cost = 3(40)+1.5(10)+12(10)+10(20)+6(10)+4(20)+50(2) +100(2) + 90(2) = RM1075.00

Generally, Table 4 shows that the feasible production plans of our heuristic method, where we need to produce 40 units of item 1, 10 units of item2 and 3 each at period 1, then produces 20 units of item 2 at period 2.in period 3, we only produce 20 units of item 3. Then lastly produces 10 units of item 1 again at period 5. Notes that, the schedule given are satisfied the condition that all external demand is meet at time. Thus, the total production cost is RM 1075.

4.1.1.4 The implementation in LINDO for model (2):

Assume that p_j as the production time need to product one unit of item *j*, which take the value p_j = 1 min, for all *j*. Assume that M is the number of machine in use, let say M=1.

This model is also solving by using the software LINDO. Where we began by opening a new blank LINDO Model Window and save it as a file. After defining a new blank LINDO Model Window, develop the objective and defining the variables of this problem with considering the model (2), which is the integer programming formulation. The implementation of the objective function and the variables is show as following:

The objective function will becomes:

(a) Minimize Z = $0.5I_{11} + 0.5I_{12} + 0.5I_{13} + 0.5I_{14} + 0.5I_{15} + 0.5I_{16} +$

$$50X_{11} + 50X_{12} + 50X_{13} + 50X_{14} + 50X_{15} + 50X_{16} +$$

$$2I_{21} + 2I_{22} + 2I_{23} + 2I_{24} + 2I_{25} + 2I_{26} +$$

$$100X_{21} + 100X_{22} + 100X_{23} + 100X_{24} + 100X_{25} + 100X_{26} +$$

$$I_{31} + I_{32} + I_{33} + I_{34} + I_{35} + I_{36} +$$

$$90X_{31} + 90X_{32} + 90X_{33} + 90X_{34} + 90X_{35} + 90X_{36}$$

where x_{jt} is the binary variable, which indicates whether a setup for item *j* occurs in period *t* for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T. It is taken the value 0 or 1, $(x_{jt} = 1)$ or not $(x_{jt} = 0)$, and I_{jt} is the inventory for item *j* at the end of period *t*, for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T.

The variable and the constraints:

From the previous section, we have assume that v_j = 0, and I_{jo} = 0, and given that $d_{12} = d_{13} = d_{14} = 10, d_{16} = 20$, $d_{21} = d_{22} = d_{25} = 10$, $d_{31} = d_{33} = d_{36} = 10$. Therefore, the equation

(b) $$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} - \sum_{i \in S_j} a_{ji} q_{it}, \qquad j = 1,...,J, \quad t = 1,...,T,$$

Becomes,

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} \qquad j = 1,...,3, \quad t = 1,...,6,$$

Thus after taken the parameter value, we will have:

(i) For *j* = 1, and *t* = 1,2...6

$$I_{11} = 0 + q_{11} - 0$$
$$I_{12} = I_{11} + q_{12} - 10$$
$$I_{13} = I_{12} + q_{13} - 10$$
$$I_{14} = I_{13} + q_{14} - 10$$
$$I_{15} = I_{14} + q_{15} - 0$$
$$I_{16} = I_{15} + q_{16} - 20$$

(ii) For j = 2, and t = 1,2...6

$$I_{21} = 0 + q_{21} - 10$$
$$I_{22} = I_{21} + q_{22} - 10$$
$$I_{23} = I_{22} + q_{23} - 0$$
$$I_{24} = I_{23} + q_{24} - 0$$
$$I_{25} = I_{24} + q_{25} - 10$$
$$I_{26} = I_{25} + q_{26} - 0$$

(iii) For j = 3, and t = 1,2...6

$$I_{31} = 0 + q_{31} - 10$$
$$I_{32} = I_{31} + q_{32} - 0$$
$$I_{33} = I_{32} + q_{33} - 10$$
$$I_{34} = I_{33} + q_{34} - 0$$
$$I_{35} = I_{34} + q_{35} - 0$$
$$I_{36} = I_{35} + q_{36} - 10$$

(C) From $p_j q_{jt} \le C_{m_j,t} y_{jt}$, $t = 1,\ldots,6,\quad j = 1,\ldots,3$, and substituted $y_{jt} = x_{jt}$, and we has assume earlier that p_2 = 5 min and p_j = 1 min, for all $j \neq 2$. Notes that $C_{m_j,t}$ is stand for the available capacity of the machine m in period t. Assume that $C_{m_j t}$ here is equal to 60. Therefore we will obtain,

$$
\begin{array}{lll}
q_{11} \le 60x_{11}, & 5q_{21} \le 60x_{21}, & q_{31} \le 60x_{31}, \\
q_{12} \le 60x_{12} & 5q_{22} \le 60x_{22} & q_{32} \le 60x_{32} \\
q_{13} \le 60x_{13}, & 5q_{23} \le 60x_{23}, & q_{33} \le 60x_{33}, \\
q_{14} \le 60x_{14}, & 5q_{24} \le 60x_{24}, & q_{34} \le 60x_{34}, \\
q_{15} \le 60x_{15}, & 5q_{25} \le 60x_{25}, & q_{35} \le 60x_{35}, \\
q_{16} \le 60x_{16} & 5q_{26} \le 60x_{26} & q_{36} \le 60x_{36}
\end{array}
$$

(d) Because we are assume that all item are produced in a same machine therefore, Γ_m here is the set of $j = 1, 2, 3$. Thus by expending equation $\sum_{j \in \Gamma_m} p_j q_{jt} \le C_{mt}$, for t = 1,..,6 and m=1. We have obtain

$$q_{11} + 5q_{21} + q_{31} \le 60$$

$$q_{12} + 5q_{22} + q_{32} \le 60$$

$$q_{13} + 5q_{23} + q_{33} \le 60$$

$$q_{14} + 5q_{24} + q_{34} \le 60$$

$$q_{15} + 5q_{25} + q_{35} \le 60$$

$$q_{16} + 5q_{26} + q_{36} \le 60$$

After consider all the parameter, it can be implemented using LINDO package.

4.1.1.5 The Result by Al Kimms Model

Using LINDO, the result of Al-Kimms model is as follows.

item \ period	1	2	3	4	5	6
1	-	10	40	-	-	-
2	10	10	-	-	10	-
3	10	-	20	-	-	-

Table 7: Quantity produced in each period

Table 7 shows the quantity produced for each items in each period. It gives the quantity $q_{12} = 10$, $q_{13} = 40$, $q_{21} = 10$, $q_{25} = 10$, $q_{31} = 10$, and $q_{33} = 20$. As before the empty space means that there is no production run in that period.

item \ period	1	2	3	4	5	6
1	-	-	30	20	20	-
2	-	-	-	-	-	-
3	-	-	10	10	10	-

Table 8: Inventory in hand

item \ period	1	2	3	4	5	6
1	-	1	1	-	-	-
2	1	1	-	-	1	-
3	1	-	-	-	-	-

Table 9: Binary Variable indicator for Setup

Table 8 show the inventory for item *j* at the end of period *t*. Thus, here we have for item 1, $I_{13} = 30$, $I_{14} = 20$, $I_{15} = 20$. There is no inventory for item 2 during all the 6 period, because all the production quantity is use to meet the demand that occur at each period. Thus, for item 3 the inventory in hand at the end of period *t* are $I_{33} = 10$, $I_{34} = 10$, and $I_{35} = 10$. The empty space means, there is no inventory at the end of the period for certain item. Moreover, Table 9 shown the binary variable indicate that which item is setup in a period. Here, $X_{12}, X_{13}, X_{21}, X_{22}, X_{25}, X_{31}$ is taken the value of 1 and the rest is 0. Thus, the total cost given by this model is:

Total cost = 0.5(30) + 0.5(20) + 0.5(20) + 1(10) + 1(10)+1(10)
+2(50)+2(90)+3(100)
= RM645.00

From the analysis, Table 7 gives a feasible schedule given by Al Kimms model. Note that, in this case the lot size schedule given is difference from the schedule given by Billington model where the production planning is produced 10 units of item 2 and 3 at period 1, 10 units each of item 1 and 2 at period 2. Then the production run at period 3 is to produce 40 units of item 1 and 20 units of item 3. There is no production run at period 4 and 6, but there are 10 units of item 2 produce in the period 5. The discussion about the comparisons of these three results will be given in the next chapter.

After studying the single level- multi items cases, we then explore the multi level, multi items capacitated lot size problem.

4.1.2 Case two: Multi level multi item capacitated lot size problem with a bottleneck

Now, consider multi-level, multi items problem. The different between the single level-multi item and multi level-multi item is that the later one has more then one level of product structure (see Figure 7).

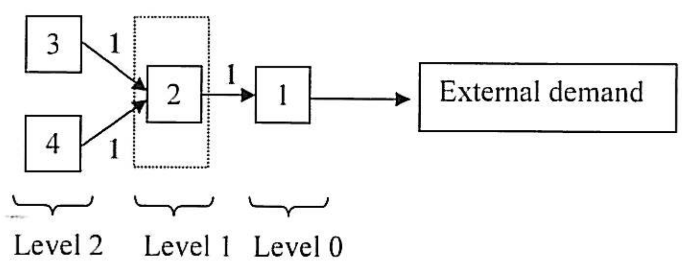


Figure 7: Multi level multi items product structure set 2

This problem has only one bottleneck facility and one item produce in the bottleneck facility. Moreover, there is only one end item, which item 1. To produce one unit of item 1, it needed input of one unit of item 2. However, to produce one unit of item 2 it needs input of one unit of item 3 and 4. The size of the items is J, let say J = 4, and T = 5, and the 'gozinto' factor for this problem is take the value, $a_{21} = 1$, $a_{32} = 1$, $a_{42} = 1$ and $a_{ji} = 0$ for others. This means that, there have the relationships for some items with some other items. For example, item1 has relationship with item 2, where we needed one unit of item 2 to produce one unit of item 1 and so on. For information, the maximum number each items $N_{\max j}$ can be produce in each period are $N_{\max 1}$ is 50, $N_{\max 2}$ is 50, $N_{\max 3}$ is 50, $N_{\max 4}$ is 50. Assume that the initial inventory for all item are assume to be zero and similarly for lead-time. And the numbers of external demand exists for end items, are: d_{12} is 10, d_{13} is 20, d_{14} is 20 and d_{15} is 10.

Furthermore, other parameters include the production time of j unit item on the bottleneck facility, b_j. For example to produce one unit of item 2 is 1 minutes then $b_2 = 1$ min. Here we assume $b_j = 1$ minutes, for $\forall j$ and the setup time, $S_j = 1$ min for $\forall j$, for the work centre for product j in the bottleneck facility.

Suppose that, we set the holding cost for item j = 1, 2, 3, 4 as 0.5, 2, 1, 3. Then we set the setup cost for item j = 1, 2, 3, 4 as 50, 100, 90, 200. And assume that the capacity utility is approximate 90 percents. Again, we assume that CAP_t is the available capacity of the work centre at time t, for t = 1, 2 ...5, so let say here we set that capacity available for period 1, CAP_1 is 50, period 2, CAP_2 is 50, period 3, CAP_3 is 50, period 4, CAP_4 is 50 and period 5, CAP_5 is 50.

Here we found that, the limitation of the capacity exists in this problem and given $v_j = 0$, as v_j is a positive and integral lead-time of item j or calls safety lead-time for product j, for $\forall j$. It satisfies the condition for a bottleneck problem that exited in a production line. This gives us a reason to suggest that this v_j is a lot-sizing problem with bottleneck.

Hence, in this problem I_{jt} is the inventory for item *j* at the end of period *t*. Thus, I_{jo} is the initial inventory for item *j*. Here, we assume that for all *j* = 1, 2 ... 4, I_{jo} is equal to zero ($I_{jo} = 0$), meaning that there is on inventory in hand remaining from the previous periods before period 1. The following section describes the implementations of the models in LINDO packages.

4.1.2.1 The implementation in LINDO for model (1):

The Objective and the variables of the Bellington model is as follows:

The objective function will becomes:

(a) Minimize Z = $2.5q_{11} + 2q_{12} + 1.5q_{13} + q_{14} + 0.5q_{15} +$

$$50X_{11} + 50X_{12} + 50X_{13} + 50X_{14} + 50X_{15} +$$
$$10q_{21} + 8_{22} + 6q_{23} + 4q_{24} + 2q_{25} +$$
$$100X_{21} + 100X_{22} + 100X_{23} + 100X_{24} + 100X_{25} +$$
$$5q_{31} + 4q_{32} + 3q_{33} + 2q_{34} + q_{35} +$$
$$90X_{31} + 90X_{32} + 90X_{33} + 90X_{34} + 90X_{35} +$$
$$15q_{41} + 12q_{42} + 9q_{43} + 6q_{44} + 3q_{45} +$$
$$200X_{41} + 200X_{42} + 200X_{43} + 200X_{44} + 200X_{45}$$

where x_{jt} is the binary variable, which indicates whether a setup for item *j* occurs in period *t* for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T. It is taken the value 0 or 1, $(x_{jt} = 1)$ or not $(x_{jt} = 0)$, and q_{jt} is the production quantity for item *j* in period *t* for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T.

The variable and the constraints:

From the previous section, we have assume that $a_{ji} = 0$, $v_j = 0$, and $I_{jo} = 0$, and given that $d_{13} = d_{14} = 20$, $d_{12} = d_{15} = 10$.

Thus ,

$$\sum_{n=1}^{5} d_{1n} = d_{1\dot{n}} = 60$$

Therefore, the equation

(b) $$\sum_{n=1}^{t} q_{j,n-v_j} - \sum_{i=1}^{t} a_{ji}q_{in}] \geq \sum_{n=1}^{t} d_{jn} - I_{j^o} \qquad j=1,2,\ldots,J \quad t=1,2,\ldots,T$$

Become,

$$q_{11} + q_{12} + q_{13} + q_{14} + q_{15} \geq d_{1\dot{n}} = 60$$

$$q_{21} + q_{22} + q_{23} + q_{24} + q_{25} - q_{11} - q_{12} - q_{13} - q_{14} - q_{15} \geq d_{2\dot{n}} = 0$$

$$q_{31} + q_{32} + q_{33} + q_{34} + q_{35} - q_{21} - q_{22} - q_{23} - q_{24} - q_{25} \geq d_{3\dot{n}} = 0$$

$$q_{41} + q_{42} + q_{43} + q_{44} + q_{45} - q_{21} - q_{22} - q_{23} - q_{24} - q_{25} \geq d_{4\dot{n}} = 0$$

(c) Since there is only item 2 product in the bottleneck facility, so from

$$\sum_{j=1}^{N} [b_j P_{jt} + s_j X_{jt}] \leq CAP_t \qquad t=1,2,\ldots,5$$

we have,

$$q_{21} + X_{21} \geq 50$$
$$q_{22} + X_{22} \geq 50$$
$$q_{23} + X_{23} \geq 50$$
$$q_{24} + X_{24} \geq 50$$
$$q_{25} + X_{25} \geq 50$$

(d) $X_{jt} = 1;$ if $\begin{cases} q_{jt} > 0 \\ 0; \text{ otherwise} \end{cases}$ $j=1,2,\ldots,J$

(4)

As indicated in the previous chapter, X_{jt} is the production indicator, where it is one kind of the fixed-change problem. Upon transformation of the model and using IP, we have the following constraints:

$q_{11} - 50X_{11} \leq 0$
$q_{12} - 50X_{12} \leq 0$
$q_{13} - 50X_{13} \leq 0$
$q_{14} - 50X_{14} \leq 0$
$q_{15} - 50X_{15} \leq 0$

$q_{21} - 50X_{21} \leq 0$
$q_{22} - 50X_{22} \leq 0$
$q_{23} - 50X_{23} \leq 0$
$q_{24} - 50X_{24} \leq 0$
$q_{25} - 50X_{25} \leq 0$

$q_{31} - 50X_{31} \leq 0$
$q_{32} - 50X_{32} \leq 0$
$q_{33} - 50X_{33} \leq 0$
$q_{34} - 50X_{34} \leq 0$
$q_{35} - 50X_{35} \leq 0$

$q_{41} - 50X_{41} \leq 0$
$q_{42} - 50X_{42} \leq 0$
$q_{43} - 50X_{43} \leq 0$
$q_{44} - 50X_{44} \leq 0$
$q_{45} - 50X_{45} \leq 0$

Results of this analysis is given in the following section.

4.1.2.2 The Result by Billington Model

Then the result given from LINDO is:

item \ period	1	2	3	4	5
1				10	50
2				11	49
3				10	50
4				10	50

Table 10: Quantity produced in each period

Table 10 shows the quantity produced for each item in each period. Notes that all items are produce at period 4 and 5 only, which are $q_{14}=10$, $q_{15}=50$, $q_{24}=11$, $q_{25}=49$, $q_{34}=10$, $q_{35}=50$, $q_{44}=10$ and $q_{45}=50$. The empty space means, there is no production run in that period for certain item.

item \ period	1	2	3	4	5	6
1					1	1
2					1	1
3					1	1
4					1	1

Table 11: Binary Variable indicator for Setup

Table 11 show the binary variable indicator for setup in each period. Here X_{14}, X_{15}, X_{24}, X_{25}, X_{34}, X_{35}, X_{44}, X_{45} is taken the value of 1 and the rest is 0 (the X_{jt} which taken the value 0 is show as the blank column). Thus the total cost is RM1337.00

We can conclude that the feasible schedule using Billington model for this problem is only to run the production at period 4 and 5. We need to produce 10 units of item1, 11 units of item 2, and 10 units of item 3 and 4 each aat period 1. Furthermore, at the period 2, we need to produce 50 units of item1, 49 units of item2 and 50 units of item 3 and 4.

4.1.2.3 The Implementations with Heuristic Method

The production structure given by Fig. 4 shown that this case is a 1-end-item problem. And the bottleneck facility is located at the second item. Therefore, we apply the heuristic that propose for the 1-end-item with non-end-item in bottleneck.

However, to solve this problem with the proposed heuristic method we need to know the demand for item 2, but we have make the assumption that the external demand only occur for the end item which is the item 1 in this case. Thus from the Fig. 4 notes that item 2 is the predecessor item for item

1 (successor item), so the demand for item 2 was the dependent demand that depend on the value of the numbers of a_{ji} P_{it} for items 1. Therefore, by using the Material requirement Planning concept we will assume that the initial demand for item 2 was $d_{22} = d_{12}$, $d_{23} = d_{13}$, $d_{24} = d_{14}$, $d_{25} = d_{15}$. Thus, d_{22} is 10, d_{23} is 20, d_{24} is 20 and d_{25} is 10.

Given that all $S_{j1} = 0$, where S_{jt} is refer to the number of stock in hand at the beginning of each period and assume that $cap_t = 50$, and assume that 1-minute is need to produce each unit of item *j*. Therefore, the implementation of the solution for this question is shown as below:

Step 1: *Period 1*

Set $P_{21} = cap_1$,

Thus we have,

$$P_{21} = 50,$$

Since $P_{21} = 50 < \sum_{t=1}^{6} d_{1t} = 60$, thus $P_{21} = 50$,

Therefore, the stock in hand for the beginning of the next periods, for item 2 is given as:

$$S_{jt} = \sum_{n=1}^{t-1} P_{jn} - d_{jn} \qquad t = 2,3,\cdots$$

Thus,

$$\begin{aligned} S_{22} &= \sum_{n=1}^{1} P_{2n} - d_{2n} \\ &= P_{21} - d_{21} \\ &= 50 - 0 = 50 \end{aligned}$$

Thus, in this period the production run produces 50 units of item. Since there is no dependent demand occur for item two in period 1 thus the 50 units of item 2 will play as the inventory in hand for the next periods to fulfil the dependent demand that occur in the next few period.

Step 2: *Periods 2:*

Set $P_{22} = cap_2$,

$\therefore P_{22} = 50$

But in this period, we have 50 units item 2 as the stock of item 2 in period 1, $S_{12} = 50$. Thus $P_{22} = 50 > \sum_{t=2}^{5} d_{2t} - S_{22} = 10$, therefore we set $P_{22} = 10$

We do this because there are wastage occurs if we produced 50 units of item 2 because the total demand for the next periods are 60 units and we still have 50 units of item 2 in hand. Therefore, it is enough to produce 10 units of item 2 to fulfil the dependent demand of item 2.

Therefore, the stock in hand for the beginning of the next periods, for item 2 is given as:

$$\begin{aligned} S_{23} &= \sum_{n=0}^{2} P_{2n} - d_{2n} \\ &= S_{22} + P_{22} - d_{22} \\ &= 50 + 10 - 10 \\ &= 50 \end{aligned}$$

In this period the production run produces 10 units of item 2. Thus, the stock in hand for item 2 is still 50 units since there is only 10 units of item 2 use to fulfil the dependent demand in this period.

Step 3: *Periods 3:*

Set $P_{23} = cap_3$,

Therefore, we have

$$P_{23} = 50,$$

Since $P_{33} > \sum_{t=3}^{5} d_{3t} - S_{33} = 0$, thus we set $P_{33} = 0$

Since the stock of item 2 in hand are enough to fulfil the total demand of item 2 in this and next periods, thus there is no production run need in this period.

Therefore, the stock in hand for the beginning of the next periods, for item 2 is given as:

$$\begin{aligned} S_{24} &= \sum_{n=1}^{3} P_{2n} - d_{2n} \\ &= S_{23} + P_{23} - d_{23} \\ &= 50 + 0 - 20 \\ &= 30 \end{aligned}$$

Since started period 3 the stock in hand already enough to fulfil the total number of dependent demand for item 2, so there is no need to run the production of item 2 in the next few periods. Thus we will have

At *period 4:*

$P_{24} = 0$,

$$\begin{aligned} S_{25} &= \sum_{n=1}^{4} P_{2n} - d_{2n} \\ &= S_{24} + P_{24} - d_{24} \\ &= 30 + 0 - 20 \\ &= 10 \end{aligned}$$

At Period 5:

$P_{25} = 0$, and there is no stock left in hand cause all have use to fulfil the dependent demand that occur in this period.

Thus, we can summary the finding that we found here as show as table below;

item \ period	1	2	3	4	5	6
1						
2	50	10				
3						
4						

Table 12: Quantity produced in each period

period / item	1	2	3	4	5	6
1						
2	1	1				
3						
4						

Table 13: Binary Variable indicator for Setup

Note that, the heuristic is only use to find the sub-optimal solution for items, which is product at the bottleneck facility. Therefore, in this case the solution is only given for item2, the rest items we will use the concept of MRP to find the solution. From Fig.5, the structures of product have 3 levels, which are level 0, level 1 and level 2. Level 0 is referring to the final product, and level 1 and 2 are referring to the parts. Thus the MRP table below showed the analysis of the assembly of item1.

Level 0- item 1	Periods				
	1	2	3	4	5
Gross requirements	-	10	20	20	10
Opening stock	-	-	-	-	-
Net requirement	-	10	20	20	10
Start assembly	-	10	20	20	10
Schedule receipts	-	10	20	20	10

Table 14 : MRP table for item 1

For item 2 we have a set of schedule that we analysed in t the previous section. Therefore we put the value into the MRP table for item 2. Thus it is shows as Table 15:

Level 1- item 2	Periods				
	1	2	3	4	5
Gross requirements	-	10	20	20	10
Opening stock	-	50	50	30	10
Net requirement	-	-	-	-	-
Start assembly	50	10	-	-	-
Schedule receipts	50	10			

Table 15 : MRP table for item 2

Once again Fig.5, shows that to produce one unit of item 2 needed one unit of item 3 and one units of item 4. These two items are at the same level, which is at level 2. Since, item 2 started to produce at period 1, so the MRP table for item 3 and 4 is as follows:

Level 2- item 3	Periods				
	1	2	3	4	5
Gross requirements	50	10	-	-	-
Opening stock	-	-	-	-	-
Net requirement	50	10	-	-	-
Start assembly	50	10			
Schedule receipts	50	10			

Table 16 : MRP table for item 3

Level 0- item 4	Periods				
	1	2	3	4	5
Gross requirements	50	10			
Opening stock	-	-			
Net requirement	50	10			
Start assembly	50	10			
Schedule receipts	10	10			

Table 17 : MRP table for item 4

Therefore, the production schedule is shows as below:

item \ period	1	2	3	4	5
1		10	20	20	10
2	50	10			
3	50	10			
4	50	10			

Table 18: Quantity produced in each period

item \ period	1	2	3	4	5
1		1	1	1	1
2	1	1			
3	1	1			
4	1	1			

Table 19 : Binary Variable indicator for Setup

With the total cost = RM2575.00, therefore, based on the solution we have, we need produce 50 units of item 2 ,3 and item 4 each at period 1. Then production 10 units of item 1 in period 2 and also 10 units of item 2, 3 and item 4 each again. After that produce 20 units of item 1 at period 3 and 4, then 10 units at periods 5 to meet the external demand for item1, since they're already have enough resources to produce item 1. And the total production cost is RM2575.

4.1.2.4 The implementation in LINDO for model (2):

Assume that p_j as the production time need to product one unit of item *j*, which take the value p_j = 1 min, for all *j*. Assume that M is the number of machine in use, let say M=1.

This model is also solving by using the software LINDO. In solving this problem we began by opening a new blank LINDO Model Window and save it as a file. After defining a new blank LINDO Model Window, develop the objective and defining the variables of this problem with considering the model (2), which is the integer programming formulation. The implementation of the objective function and the variables is show as following:

The objective function will becomes:

(b) Minimize Z =

$$\begin{aligned}
&0.5I_{11} + 0.5I_{12} + 0.5I_{13} + 0.5I_{14} + 0.5I_{15} + \\
&50X_{11} + 50X_{12} + 50X_{13} + 50X_{14} + 50X_{15} + \\
&2I_{21} + 2I_{22} + 2I_{23} + 2I_{24} + 2I_{25} + \\
&100X_{21} + 100X_{22} + 100X_{23} + 100X_{24} + 100X_{25} + \\
&I_{31} + I_{32} + I_{33} + I_{34} + I_{35} + \\
&90X_{31} + 90X_{32} + 90X_{33} + 90X_{34} + 90X_{35} + \\
&3I_{41} + 3I_{42} + 3I_{43} + 3I_{44} + 3I_{45} + \\
&200X_{41} + 200X_{42} + 200X_{43} + 200X_{44} + 200X_{45}
\end{aligned}$$

Where x_{jt} is the binary variable, which indicates whether a setup for item *j* occurs in period *t* for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T. It is taken the value 0 or 1, $(x_{jt} = 1)$ or not $(x_{jt} = 0)$, and I_{jt} is the inventory for item *j* at the end of period *t*, for all *j* = 1, 2 ... *J* and *t* = 1, 2 ...T.

The variable and the constraints:

From the previous section, we have assume that $v_j = 0$, and $I_{jo} = 0$, and given that $d_{13} = d_{14} = 20$, $d_{12} = d_{15} = 10$ and T = 5, *J* = 4. Therefore, the equation

(b) $$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} - \sum_{i \in S_j} a_{ji} q_{it}, \qquad j = 1,...,J, \qquad t = 1,...,T,$$

Thus after taken the parameter value, then we will have:

(i) For *j* = 1, and *t* = 1,2...5

$$I_{11} = 0 + q_{11} - 0$$
$$I_{12} = I_{11} + q_{12} - 10$$
$$I_{13} = I_{12} + q_{13} - 20$$
$$I_{14} = I_{13} + q_{14} - 20$$
$$I_{15} = I_{14} + q_{15} - 10$$

(ii) For *j* = 2, and *t* = 1,2...5

$$I_{21} = 0 + q_{21} - q_{11}$$
$$I_{22} = I_{21} + q_{22} - q_{12}$$
$$I_{23} = I_{22} + q_{23} - q_{13}$$
$$I_{24} = I_{23} + q_{24} - q_{14}$$
$$I_{25} = I_{24} + q_{25} - q_{15}$$

(iii) For *j* = 3, and *t* = 1,2...5

$$I_{31} = 0 + q_{31} - q_{21}$$
$$I_{32} = I_{31} + q_{32} - q_{22}$$
$$I_{33} = I_{32} + q_{33} - q_{23}$$
$$I_{34} = I_{33} + q_{34} - q_{24}$$
$$I_{35} = I_{34} + q_{35} - q_{25}$$

(iv) For j = 4, and t = 1,2…5

$$I_{41} = 0 + q_{41} - q_{21}$$
$$I_{42} = I_{41} + q_{42} - q_{22}$$
$$I_{43} = I_{42} + q_{43} - q_{23}$$
$$I_{44} = I_{43} + q_{44} - q_{24}$$
$$I_{45} = I_{44} + q_{45} - q_{25}$$

(C) From $p_j q_{jt} \le C_{m_j t} y_{jt}$, $t = 1,\dots,6,\quad j = 1,\dots,3$,

and substituted $y_{jt} = x_{jt}$, and we has assume earlier that p_j = 1 min, for all j. Therefore we will obtain,

$q_{11} \le 50x_{11}$,	$q_{21} \le 50x_{21}$,	$q_{31} \le 50x_{31}$,	$q_{41} \le 50x_{41}$,
$q_{12} \le 50x_{12}$	$q_{22} \le 50x_{22}$	$q_{32} \le 50x_{32}$	$q_{42} \le 50x_{42}$
$q_{13} \le 50x_{13}$,	$q_{23} \le 50x_{23}$,	$q_{33} \le 50x_{33}$,	$q_{43} \le 50x_{43}$,
$q_{14} \le 50x_{14,}$	$q_{24} \le 50x_{24,}$	$q_{34} \le 50x_{34,}$	$q_{44} \le 50x_{44,}$
$q_{15} \le 50x_{15}$,	$q_{25} \le 50x_{25}$,	$q_{35} \le 50x_{35}$,	$q_{45} \le 50x_{45}$

(d) After that expending equation $\sum_{j\in\Gamma_m} p_j q_{jt} \le C_{mt}$, for t = 1…5 and m=1.

Thus we have

$$q_{11} + q_{21} + q_{31} + q_{41} \le C_{11}$$
$$q_{12} + q_{22} + q_{32} + q_{42} \le C_{12}$$
$$q_{13} + q_{23} + q_{33} + q_{43} \le C_{13}$$
$$q_{14} + q_{24} + q_{34} + q_{34} \le C_{14}$$
$$q_{15} + 5q_{25} + q_{35} + q_{45} \le C_{15}$$

After transformation, it can be implemented using LINDO, we have the following result which is similar to the previous section.

4.1.2.4.1 The Result by Al Kimms model

Then the result given from LINDO is :

item \ period	1	2	3	4	5
1	-	16	17	17	10
2	-	17	16	17	10
3	--	17	17	16	10
4	50	-	-	-	10

Table 20 : Quantity produced in each period

Table 20 show the quantity produced for each items in each period. It give that for item 1 the production run is to $q_{12} = 16$, $q_{13} = 17$, $q_{14} = 17$, $q_{15} = 10$. Thus for item 2, $q_{22} = 17$, $q_{23} = 16$, $q_{24} = 17$ and $q_{25} = 10$. Then for item 3 $q_{32} = 17$, $q_{33} = 17$, $q_{34} = 17$ and $q_{35} = 10$. Moreover, there are only two production run to produce item 4, which is $q_{41} = 50$, and $q_{45} = 10$. The empty space means, there is no production run in that period for certain item.

item \ period	1	2	3	4	5
1	-	6	3	-	-
2	-	1	-	-	-
3	-	-	1	-	-
4	50	33	17	-	-

Table 21 : Inventory in hand

item \ period	1	2	3	4	5
1	-	1	1	1	1
2	-	1	1	1	1
3	-	1	1	1	1
4	1	-	-	-	1

Table 22: Binary Variable indicator for Setup

Table 21 show the inventory for item j at the end of period t. Thus, here we have for item 1, $I_{12} = 6$ and $I_{13} = 3$. There is only one units of item 2 left, $I_{22} = 1$ at the end of the period 2 during the entire time horizon. This situation also occurs for item 3 but it is at period 3, $I_{33} = 1$. For item 4, $I_{41} = 50$, $I_{42} = 33$, $I_{43} = 17$. The empty space means, there is no inventory at the end of the period for certain item. Moreover, Table 8 shown the binary variable indicate that which item is setup in a period. Here, X_{12}, X_{13}, X_{14}, X_{15}, X_{22}, X_{23}, X_{24}, X_{25}, X_{32}, X_{33}, X_{34}, X_{35}, and X_{42}, X_{43}, X_{54}, X_{45}, is taken the value of 1 and the rest is 0. Thus, the total cost given by this model is:

Total cost = 6(0.5) + 3(0.5)+ 1(2) + 1 + 50(3) + 33(3) +17(3)+ 4(50) + 4(100) + 4(90) + 2(200) = RM1667.50

Thus, as conclusion the production plan that we get from this analyse is shown at Table 20, where the schedule shown that we need to produce 50 units of item 4 at period 1. Follow by, 16 units of item 1, 17 units of item 2 and 3 at period 2. Next, produce 17 units of item1 and 3 and 16 units of item 2 at period 3. Then 17 units of item 1 and 2 each and 16 units item 3 at period 4. At period 5, still have 10 units of item need to produce to meet the external demand that occur at that period. Since the capacity are available, thus all item 2, 3 and 4 will produce to produce item 1 at the same period.

4.2 Summary

Generally, the answer given by each method is different from each other. For the first case, the total cost was RM754 with method devised using Billington while RM 1075 and RM645 were the cost given by our heuristic method and Alf Kimms method. We conclude that the total production cost that obtain by method proposes by Alf Kimms is the lowest compare to the other two methods for case one. Even though our model does not give the best result, it gives a much simpler model and it does gave an acceptable feasible solution.

Chapter 5

Discussion and Suggestion

5.0 Heuristic model for Billington and Alf Kimms Model.

In this study we worked on two basic models that were used to solve multi-level lot-sizing problem namely the Billington and Alf Kimms model. We proposed a new heuristic model based on integer programming to solve the Billington model. A few cases were introduced to demonstrate how the system works and the results from the analysis as described in chapter 4.

Generally we found that, the Billington model is not suitable to be used directly using LINDO but require the support from a heuristic method. The result generated using Billington alone are normally not feasible. Table 1 shows the production plan and note that the production runs starts at period 5. However, the external demand for item one occurs since period 2. The solution given failed to fulfil the external demand exactly at the period that the demand occurs. Suppose the external demand for item 1 only occurs at period 2, where $d_{12} = 50$. The rest of the parameters remain the same as case 1 described in chapter 4. Thus, by using the Alf Kimms model, we will obtain a new set of result as follow:

item \ period	1	2	3	4	5	6
1	-	-	10	40	-	-
2	10	10	-	-	10	-
3	10	-	20	-	-	-

Table 23 : Quantity produced in each period

With the total production cost RM635. Note that the schedules given are already difference from the pervious one (Table 7), although the total sum of the demand is still the same. However, we will set the same schedule as what we have at Table 1 for this case if we solve it by Billington model. This is due to the same reason described earlier. That is, in the model Billington, the constraint (2) (see Chapter 3) that illustrates available production consider the total sum of all external demands for each end item that occur during the horizon of T periods. Since the total external demand occur for this case equals to 50 units, thus it is still the same for the previous case we have the same feasible schedule like before, if we solve problem by using Billington model.

Therefore, a simple heuristic is developed to solve this problem. In Billington et al (1986), a heuristic method based on the Lagrangian relaxation with a branch and bound procedure was introduced. As we have mentioned earlier chapters, in this study we are consider a new heuristic. This method will only give us a feasible sub-optimal solution so as Alf Kimms model. However, comparing the result given by this two analysis, the later one gave a more efficient solution, where it has the lowest cost compare with the heuristic method.

Even though the cost generated using our new heuristic is more expensive than Alf Kimm's model, our feasible schedules can be reduced by considering the sequence of the setup item for producing a particular item. However, our heuristic method is more feasible for the single-level item problem. Consider again case 1 in chapter 4. The feasible schedule given by our heuristic method is show as following:

item \ period	1	2	3	4	5	6
1	40	-	-	10	-	-
2	10	20	-	-	-	-
3	10	-	20	-	-	-

Table 24 : Quantity produced in each period

Note that the schedule in the table shows that which item and the number of items will be produced in each period. For each problem, example the problem that it have the assembly production structure, we will know that the predecessor item are required to product in a particular period. However, the question that which item will produced first was became another problems that we need to solve and observations that it will affected the total production cost in term of the setup cost or some time is in term of holding cost (this is another cases). Let consider again the table above, the table show that we need to produce item 1, 2 and 3 in the period 1. Since there are no precedence relationship among these three item, we continuous work on the feasible schedule given, so that we will have a more efficient feasible schedule that will reduce the cost that we found before that. Note that, in this case item 2 will continuous to product at period 2. Thus if we schedule that at period 1 we first produce item 1 or 3 first then produce item 2 at the end of the period 1. Then continue to produce item 2 at the beginning of the period 2. By this schedule we will save the setup cost that cost when there is a new production run is setup for certain item. Hence, the total setup cost for item 2 at period 2 can excluding from the the calculation of total cost.

In the real world situation, there are many items produced in a certain period. There are many possible feasible sub-optimal solutions that we can obtain. Thus, for further study, we think lie in further studies of the multi-level lot-sizing problem with consider the sequence of the production item. Since, the sequence of production item is also one kind of the factor that will affected our objective in solving this problem which is to find out a set of feasible solution that will minimize our total production setup cost and holding cost. And these works can carry out by using the Simulated Annealing or Genetic Algorithm heuristic.

BIBLIOGRAPHY

Arnold Kaufmann & Arnaud Herry-Laborděre (1977). "Integer and Mixed Programming; Theory and Application." New York: Academic Press.

Barry Render, Ralph M. Stair Jr. & Michael E. Hanna (2003). "Quantitative Analysis For Management." 8th edition. New Jersey: Prentice Hall

Fleischmann B (1994). "The Discreet Lot Sizing and Scheduling Problem with Sequence-Dependent Setup-Costs." European Journal of Operational Research. 75. 395-404.

Hamdy A. Taha (2003). "Operation Research: An introduction." 7th Edition. New Jersey: Prentice Hall

Hyun C J, Kim Y & Kim Y K (1998). "A Genetic Algorithm For Multiple Objective Sequencing Problems In Mixed Model Assembly Lines." Computers and Operations Research. Vol.25, No. 7/8. 675-690.

Khouja M, Michalewicz Z & Wilmot M (1998). "The Use Of Genetic Algorithms To Solve The Economic Lot Size Scheduling Problem." European Journal of Operational Research 110. 509-524.

Kim J & Kim Y (1996). "Simulated Annealing and Genetic Algorithms for Scheduling Products with Multi-level Product Structure." Computers Operations Research. Vol.23, No. 9. 857-868.

Mordecai Avriel & Boaz Golany (1996). "Mathematical Programming for Industrial Engineers." New York: Marcel Dekker

Wolsey L A (1997). "MIP Modelling of Changeovers in Production Planning and Scheduling Problems." European Journal of Operational Research 99. 154-161.

APPENDIX A

```
MIN   3Q11+2.5Q12+2Q13+1.5Q14+Q15+0.5Q16+12Q21+10Q22+8Q23+
      6Q24+4Q25+2Q26+  6Q31+5Q32+4Q33+3Q34+2Q35+Q36+
      50x11+50x12+50x13+50x14+50x15+50x16+100x21+100x22+100x23+
      100x24+100x25+100x26+90x31+90x32+90x33+90x34+90x35+90x36

Subject to

Q11+Q12+Q13+Q14+Q15+Q16=>50
Q21+Q22+Q23+Q24+Q25+Q26=>30
Q31+Q32+Q33+Q34+Q35+Q36=>30
Q11+5Q21+Q31+x11+x21+x31<=60
Q12+5Q22+Q32+x12+x22+x32<=60
Q13+5Q23+Q33+x13+x23+x33<=60
Q14+5Q24+Q34+x14+x24+x34<=60
Q15+5Q25+Q35+x15+x25+x35<=60
Q16+5Q26+Q36+x16+x26+x36<=60

Q11-60x11<=0
Q12-60x12<=0
Q13-60x13<=0
Q14-60x14<=0
Q15-60x15<=0
Q16-60x16<=0

Q21-60x21<=0
Q22-60x22<=0
Q23-60x23<=0
Q24-60x24<=0
Q25-60x25<=0
Q26-60x26<=0

Q31-60x31<=0
Q32-60x32<=0
Q33-60x33<=0
Q34-60x34<=0
Q35-60x35<=0
Q36-60x36<=0

END
GIN 18

INTE x11
INTE x12
INTE x13
INTE x14
INTE x15
INTE x16
INTE x21
INTE x22
INTE x23
INTE x24
INTE x25
INTE x26
INTE x31
INTE x32
INTE x33
INTE x34
INTE x35
INTE x36
```

APPENDIX B

```
MIN  0.5I11+0.5I12+0.5I13+0.5I14+0.5I15+0.5I16+2I21+2I22+2I23+
     2I24+2I25+2I26+I31+I32+I33+I34+I35+I36+
     50X11+50X12+50X13+50X14+50X15+50X16+100X21+100X22+100X23+
     100X24+100X25+100X26+90X31+90X32+90X33+90X34+90X35+90X36

Subject to

I11-Q11= 0
I12-I11-Q12= -10
I13-I12-Q13= -10
I14-I13-Q14= -10
I15-I14-Q15= 0
I16-I15-Q16= -20

I21-Q21= -10
I22-I21-Q22= -10
I23-I22-Q23= 0
I24-I23-Q24= 0
I25-I24-Q25= -10
I26-I25-Q26= 0

I31-Q31= -10
I32-I31-Q32= 0
I33-I32-Q33= -10
I34-I33-Q34= 0
I35-I34-Q35= 0
I36-I35-Q36= -10

Q11+5Q21+Q31<=60
Q12+5Q22+Q32<=60
Q13+5Q23+Q33<=60
Q14+5Q24+Q34<=60
Q15+5Q25+Q35<=60
Q16+5Q26+Q36<=60

5Q21-60X21<=0
5Q22-60X22<=0
5Q23-60X23<=0
5Q24-60X24<=0
5Q25-60X25<=0
5Q26-60X26<=0

Q11-60X11<=0
Q12-60X12<=0
Q13-60X13<=0
Q14-60X14<=0
Q15-60X15<=0
Q16-60X16<=0

Q31-60X31<=0
Q32-60X32<=0
Q33-60X33<=0
Q34-60X34<=0
Q35-60X35<=0
Q36-60X36<=0

END

GIN 18

INTE X11
INTE X12
INTE X13
INTE X14
INTE X15
INTE X16
INTE X21
INTE X22
INTE X23
INTE X24
INTE X25
INTE X26
INTE X31
INTE X32
INTE X33
INTE X34
INTE X35
INTE X36
```

APPENDIX C

```
MIN 2.5Q11+2Q12+1.5Q13+Q14+0.5Q15+10Q21+8Q22+6Q23+4Q24+2Q25+
    5Q31+4Q32+3Q33+2Q34+Q35+15Q41+12Q42+9Q43+6Q44+3Q45+
    50x11+50x12+50x13+50x14+50x15+100x21+100x22+100x23+100x24+
    100x25+90x31+90x32+90x33+90x34+90x35+200x41+200x42+200x43+
    200x44+200x45

Subject to
Q11+Q12+Q13+Q14+Q15=>60
Q21+Q22+Q23+Q24+Q25-Q11-Q12-Q13-Q14-Q15=>0
Q31+Q32+Q33+Q34+Q35-Q21-Q22-Q23-Q24-Q25=>0
Q41+Q42+Q43+Q44+Q45-Q21-Q22-Q23-Q24-Q25=>0

Q21+x21<=50
Q22+x22<=50
Q23+x23<=50
Q24+x24<=50
Q25+x25<=50

Q11-50x11<=0
Q12-50x12<=0
Q13-50x13<=0
Q14-50x14<=0
Q15-50x15<=0
Q21-50x21<=0
Q22-50x22<=0
Q23-50x23<=0
Q24-50x24<=0
Q25-50x25<=0
Q31-50x31<=0
Q32-50x32<=0
Q33-50x33<=0
Q34-50x34<=0
Q35-50x35<=0
Q41-50x41<=0
Q42-50x42<=0
Q43-50x43<=0
Q44-50x44<=0
Q45-50x45<=0

END
GIN 20

INTE x11
INTE x12
INTE x13
INTE x14
INTE x15
INTE x21
INTE x22
INTE x23
INTE x24
INTE x25
INTE x31
INTE x32
INTE x33
INTE x34
INTE x35
INTE x41
INTE x42
INTE x43
INTE x44
INTE x45
```

APPENDIX D

```
MIN 0.5I11+0.5I12+0.5I13+0.5I14+0.5I15+2I21+2I22+2I23+2I24+2I25+
    I31+I32+I33+I34+I35+3I41+3I42+3I43+3I44+3I45+
    50X11+50X12+50X13+50X14+50X15+100X21+100X22+100X23+
    100X24+100X25+90X31+90X32+90X33+90X34+90X35+200X41+
    200X42+200X43+200X44+200X45

Subject to:

I11-Q11=0
I12-I11-Q12=-10
I13-I12-Q13=-20
I14-I13-Q14=-20
I15-I14-Q15=-10

I21-Q21+Q11=0
I22-I21-Q22+Q12=0
I23-I22-Q23+Q13=0
I24-I23-Q24+Q14=0
I25-I24-Q25+Q15=0

I31-Q31+Q21=0
I32-I31-Q32+Q22=0
I33-I32-Q33+Q23=0
I34-I33-Q34+Q24=0
I35-I34-Q35+Q25=0

I41-Q41+Q21=0
I42-I41-Q42+Q22=0
I43-I42-Q43+Q23=0
I44-I43-Q44+Q24=0
I45-I44-Q45+Q25=0

Q11+Q21+Q31+Q41<=50
Q12+Q22+Q32+Q42<=50
Q13+Q23+Q33+Q43<=50
Q14+Q24+Q34+Q44<=50
Q15+Q25+Q35+Q45<=50

Q11-50X11<=0
Q12-50X12<=0
Q13-50X13<=0
Q14-50X14<=0
Q15-50X15<=0

Q21-50X21<=0
Q22-50X22<=0
Q23-50X23<=0
Q24-50X24<=0
Q25-50X25<=0

Q31-50X31<=0
Q32-50X32<=0
Q33-50X33<=0
Q34-50X34<=0
Q35-50X35<=0

Q41-50X41<=0
Q42-50X42<=0
Q43-50X43<=0
Q44-50X44<=0
Q45-50X45<=0

END

GIN 20

INTE X11
INTE X12
INTE X13
INTE X14
INTE X15
INTE X21
INTE X22
INTE X23
```

INTE X24
INTE X25
INTE X31
INTE X32
INTE X33
INTE X34
INTE X35
INTE X41
INTE X42
INTE X43
INTE X44
INTE X45

APPENDIX E

The implementation in LINDO:

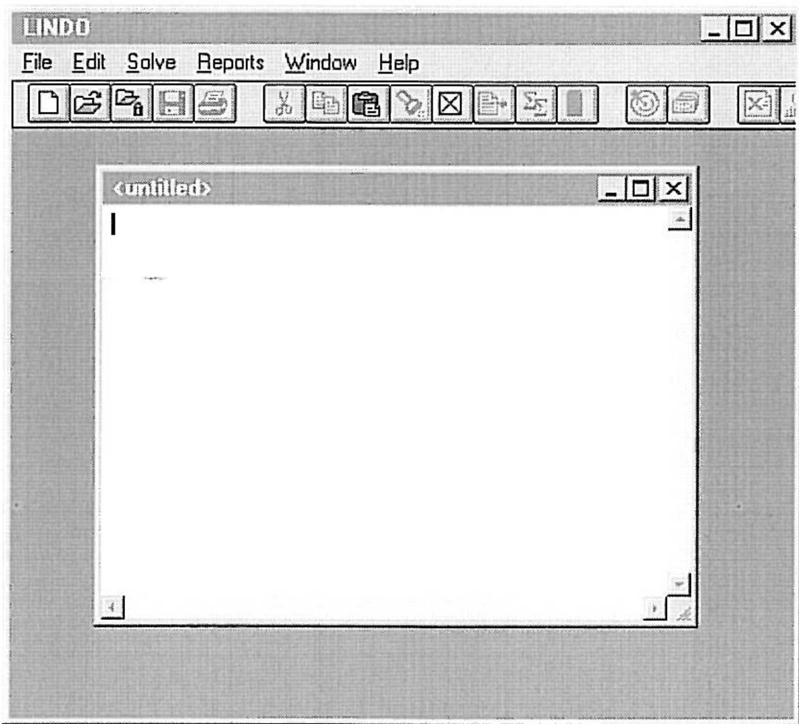
Entering a model

In developing a LINDO model, there includes the following 5 steps:

1. Opening a new blank Model Window
2. Developing an Objective & Defining the Variables
3. Determining the Constraints
4. Solving the model
5. Interpreting results in the Reports Window

1. Opening a New Blank Window

When you start LINDO, your screen will resemble the following:



[he smaller child window labeled "<untitled>" is a new, blank Model Window. We will ype our sample model directly into this window. The next step is to Develop an)bjective and Define the Variables.

!. Developing an Objective & Defining the Variables

A LINDO model has a minimum requirement of three things. It needs an)bjective, variables, and constraints.

The first requirement, an objective, is just what it sounds like: a goal. we have the :hoice of two goals, MAX or MIN, which stand for maximize and minimize The first vord in a LINDO model must be either MAX or MIN.

The formula you enter after the MAX or MIN is called the objective function. Example, XYZ wants to maximize profit achievable with the limited labor and)roduction facilities available.

Let STD and DLX be the variables, which are things that we want LINDO to idjust to reach the maximum. Where STD has a profit contribution of 10 and DLX has a)rofit contribution of 15, and then we enter it in the formula. E.g. enter:

MAX 10 STD + 15 DLX

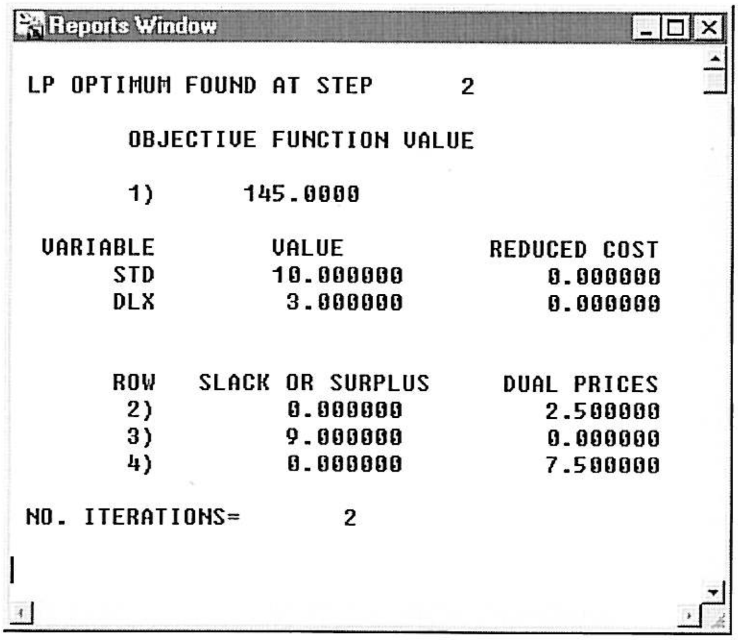
[he next step is determining the Constraints.

3. Determining the Constraints

Constraints are the most important part of our model. Example we're considering, f we were to maximize 10 STD + 15 DLX now, there's no limit to how many Standard ınd Deluxe computers we could produce. Of course, there must be some limit. In this example, this is the factory output and the labor supply. So, let's constrain both STD and DLX to be less than the factory capacity available per day of

5. Interpreting results in the Reports Window

There will now be a new window on our screen titled "Reports Window". In this window LINDO will show the report output. Getting back to the example, the Reports Window now contains the solution to the model and should resemble the following:



```
Reports Window

LP OPTIMUM FOUND AT STEP       2

        OBJECTIVE FUNCTION VALUE

        1)        145.0000

 VARIABLE          VALUE          REDUCED COST
      STD        10.000000            0.000000
      DLX         3.000000            0.000000


      ROW   SLACK OR SURPLUS       DUAL PRICES
       2)         0.000000            2.500000
       3)         9.000000            0.000000
       4)         0.000000            7.500000

NO. ITERATIONS=        2
```

Figure above shows that LINDO took 2 iterations to solve the model, the maximum profit attainable from the two variables which have constrained is 145 and the variables STD and DLX take the values 10 and 3.