MULTIPLE PLANTS CAPACITATED LOT SIZING AND SCHEDULING WITH SEQUENCE-DEPENDENT SETUP COSTS

by

Sari Hilmi Abdullah

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Engineering Systems Management

Sharjah, United Arab Emirates

November 2016

Approval Signatures

We, the undersigned, approve the Master's Thesis of Sari Hilmi Abdullah.

Thesis Title: Multiple Plants Capacitated Lot Sizing and Scheduling with Sequence-Dependent Setup Costs.

Signature	**Date of Signature** (dd/mm/yyyy)
___________________________ Dr. Abdulrahim Shamayleh, Assistant Professor, Department of Industrial Engineering Thesis Advisor	_______________
___________________________ Dr. Malick Ndiaye Associate Professor, Department of Industrial Engineering Thesis Co-Advisor	_______________
___________________________ Dr. Rami Asa'd Assistant Professor, Department of Industrial Engineering Thesis Committee Member	_______________
___________________________ Dr. Norita Ahmad Associate Professor, Department of Marketing and Information Systems, School of Business Administration, Thesis Committee Member	_______________
___________________________ Dr. Moncer Hariga Director, Engineering Systems Management Graduate Program	_______________
___________________________ Dr. Mohamed El-Tarhuni Associate Dean, College of Engineering	_______________
___________________________ Dr. Richard T. Schoephoerster Dean, College of Engineering	_______________
___________________________ Dr. Khaled Assaleh Interim Vice Provost for Research and Graduate Studies	_______________

Acknowledgements

First of all, I am thankful and grateful to Allah for giving me the patience, courage and strength to reach the aims of this research.

I would like to thank the American University of Sharjah for granting me the Graduate Teaching Assistantship Scholarship to achieve my dreams in studying one of the best graduate programs in the United Arab Emirates. My special thanks also go to the faculty of the Department of Industrial Engineering, as well as the Engineering Systems Management (ESM) program for their endless support through providing me with a high standard educational environment.

I would like to express my gratitude and appreciation to my advisors Dr. Abdulrahim Shamayleh and Dr. Malick Ndiaye for their valuable feedback, effort and guidance they provided me with; it would not have been possible without it to complete this research.

Many thanks also to the committee members Dr. Rami As'ad and Dr. Norita Ahmad for their valuable suggestions and insightful comments.

Also, I would like to thank Engineer Abdalmonam Zahed and Engineer Adnan Al Sousy for their priceless help through this research.

Finally, I would like to thank Dr. Moncer Hariga for his limitless care and support for every student in the department.

Dedication

To the most precious gift in the life, my beloved parents for their endless love, care, faith and support.

To my professors Dr. Abdulrahim and Dr. Malick the most remarkable professors.

To my friends Mustafa, Samer and Hossein for being there whenever I needed them.

Abstract

Production planning is a crucial activity for companies to satisfy customers demand while minimizing cost. The objective of this research is to optimize the production planning and scheduling decisions of companies in petrochemical industry field. A Mixed-Integer Linear Programming (MILP) model is developed for the capacitated lot sizing and scheduling problem with sequence dependent setup costs; that considers the chain of multiple suppliers, affiliates, warehouses and customers. Different grades can be produced by each affiliate, and a changeover cost occurs when changing the production from one grade to another. The model integrates scheduling and lot-sizing decisions with the logistics functions of transportation and warehousing. In particular, it provides answers to questions regarding the amount of each raw material to be purchased from each supplier, sequence of production plans, inventory levels, and warehouse selection to satisfy orders. Petrochemical companies usually own several joint-ventures and centrally prepare the affiliates' production plans for the upcoming periods; their current planning procedures do not consider the overall costs in their supply-chain. The developed model will integrate their raw material costs, production and sequencing costs, inventory costs and transportation costs from the supplier side across the supply-chain to the customer side. The problem under study is considered an NP-Hard problem due to its complexity and size; therefore, a three stage heuristic was developed which provided good quality solutions with an acceptable computational time. The first stage of the heuristic works on reducing the complexity of the model by removing the sequencing decisions; where the resulting model decides only the size of lots for each affiliate. In the second and third stages, an iterative process is done to reduce the total setup and total holding costs while restoring the sequencing decisions. The heuristic was applied on different problem instances of different sizes and the reported results were within a range of 0.09% - 2.0% away from optimality. The development of the Mixed-Integer Linear Programming and the Three Stage Heuristic is the main contribution of this research.

Search Terms: capacitated lot-sizing; sequence-dependent changeovers; mixed-integer linear programming; sequencing; production; scheduling heuristics

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AC - Ant Colony

CLSP - Capacitated Lot-sizing and Scheduling Problem

CLP - Capacitated Lot-sizing Problem

CT - Computational Time

DLSP - Discrete Lot-sizing Problem

DP - Dynamic Programming

ELSP - Economic Lot-sizing Problem

FO - Fix and Optimize

GLSP - General Lot-sizing Problem

GA - Genetic Algorithms

GH - Greedy heuristics

H - Heuristics

HA - Hierarchical Approach

LR - Lagrangean Relaxation Heuristics

MP - Mathematical Programming Heuristics

MH - Metaheuristics

PLSP - Proportional Lot-sizing Problem

SA - Simulated Annealing

Chapter 1: Introduction

The global market is becoming more competitive due to many factors such as increase of customer focus and shorter product life cycle. Therefore companies should be aware that production planning is considered as an important activity that affects the organization's performance and market position. According to Lu et al. [1], there are three levels of planning: Strategic Planning (Long Term), Tactical Planning (Medium Term) and Operational Planning (Short Term). The focus of this research is on the tactical planning level that involves decisions that are most relevant to raw material requirement planning and lot sizing over a certain period of time in order to satisfy the demand without exceeding the available capacity while incurring the lowest total costs. These decisions are vital and challenging at the same time due to the complexity that it can reach. Lot-sizing is about deciding the production orders or lots to satisfy the customers demand while considering the minimum cost [2]. Lot sizing can have different configurations in terms of classes and properties. In addition to that, several solution approaches such as mathematical heuristics and metaheuristics are used to solve the lot sizing problems in less time and effort in comparison to the commercial software with an acceptable quality solution.

1.1 Characteristics of Lot-Sizing Problems

Lot-sizing problems vary in their complexity depending on the industry's application, where the application impacts the features considered in the model [2]; therefore, they differ in terms of specifications and features. The most important lot-sizing problem characteristics are summarized by Hacer et al. [3]. Table 1 shows the summary of lot-sizing problem characteristics.

Table 1 Lot-Sizing Problems

Lot-Sizing Problems	Number of Manufacturing Levels	Single Level
		Multiple Level
	Capacity Constraints	Limited Resources
		Unlimited Resources
	Setup Time	Simple Setup
		Complex Setup

Table 1 Lot-Sizing Problems (Continued)

Lot-Sizing Problems	Planning Horizon	Infinite
		Finite
		Rolling
	Demand Type	Deterministic
		Stochastic
	Inventory Shortages	With Shortages
		Without Shortages

1. Number of Manufacturing Levels:

Production systems are often categorized as either single or multiple levels. In single level systems, the raw materials are transformed into a final product with a single operation process, while in multiple level system, the raw materials need more than one operations in order to be transferred to a final product. Figure 1 illustrates the difference between single and multiple manufacturing levels.

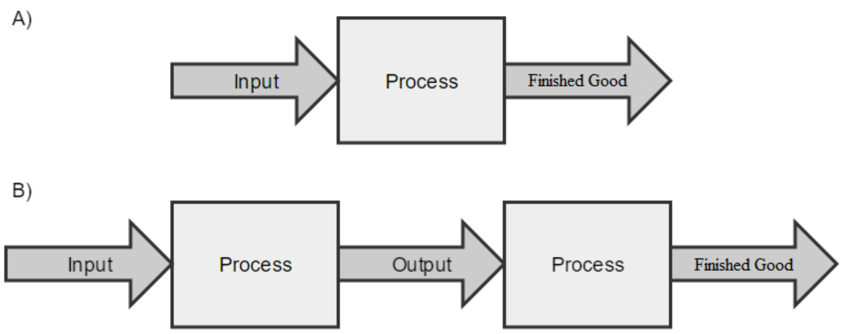

Figure 1 A) Single Level Manufacturing, B) Multiple Level Manufacturing

In Figure 1 A, only one process is needed to transform the raw material into finished goods, while in Figure 1 B, more than one processes are needed to transform the raw material into finished goods. Single level products demand can come directly from customers or from the forecasts. This type of demand is called Independent. On the other hand, in multiple level products, the demand for the components that are needed to produce the final product depends on the required quantity of the final product. This relation influences the raw material requirements and it is presented in the bill of material, therefore it should be taken into consideration while planning for the raw material requirements [3].

2. Capacity Constraints:

Capacity can be identified as the resources availability or the production capability. If there are no limits on the resources, the lot-sizing problem is called uncapacitated problem where complexity will be less than capacitated problem [3].

3. Setup Time:

Setup time/cost occurs when a production changeover happens to produce different products. The setup can be either simple or complex. The setup is considered simple if it does not depend on the sequences of the products decisions. In other words, decisions in the previous period or the coming period do not affect the current period decision. On the other hand, the complex setup structure consists of three types. The first type is setup carryover that involves continuation with the production of the same product in the next period without additional setup time or cost. Figure 2 illustrates the carry-over setup.

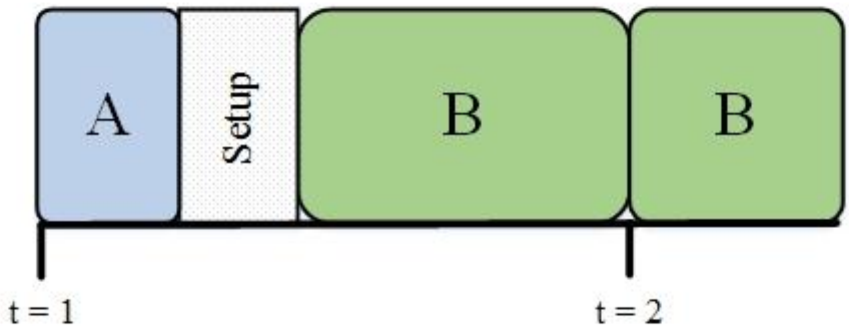

Figure 2 Setup Carry-over

From Figure 2, we can see that the last product produced at period one is product B. At the second period the same product B is being produced, therefore there is no need to have a setup between these two periods since the setup for product B is preserved. The second type is family or major setup that happens when other products from a different product family are produced at the same period. The third type is sequence-dependent setup where production sequences influence the production sequence setup cost [3]. In Figure 3 A, the setup cost of switching from product A to product B to product C is not equal to the setup cost of switching from product A to product C to product B (Figure 3 B). Even though the same products are being produced, the setup cost of switching from product A to product B is not equal to the setup cost of switching from product A to product C.

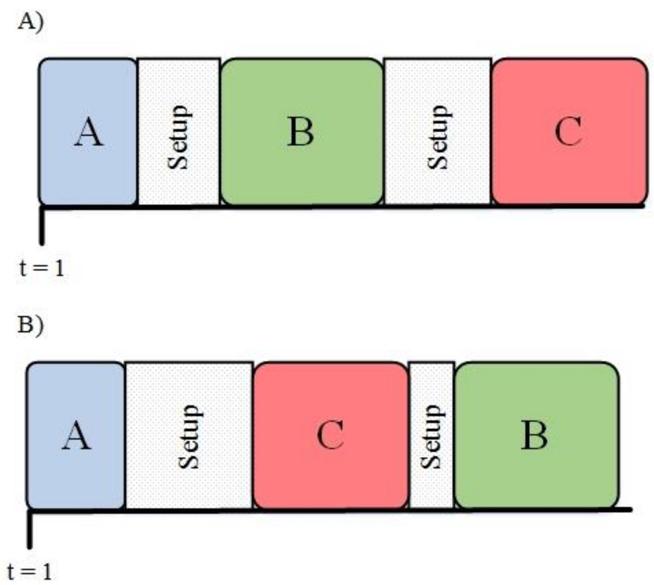

Figure 3 Sequence-Dependent Setup Cost Example

4. Planning Horizon:

Planning horizon is the time interval that the production master plan schedule into the future. The planning horizon can be one of the three types, infinite, finite and rolling. Infinite planning horizon is used usually when the demand for the products is constant, while finite planning horizon is used when the demand for the products is dynamic. In the rolling planning horizon, the planning is prepared for a certain number of periods where the demand is known, the production decisions are applied for the first period only based on the prepared plan, then the plan will be updated by increasing the length of the horizon by one additional period [3].

5. Demand Type:

Demand can be considered as either deterministic, stochastic or fuzzy. Deterministic demand is where the value of demand is known with high confidence. Deterministic demand can be static or dynamic. Stochastic demand is defined as demand with level of uncertainty. When new products are launched and demand cannot be predicted due to the lack of historical data, the demand can be described as low or high or approximated based on subjective experts opinion, this is referred as fuzzy demand [3].

6. Inventory Shortages:

Inventory shortage or backlogging is a feature where the demand for the current period can be satisfied by the next periods [3].

1.2 Classes of Lot-Sizing and Scheduling Problems

Lot-sizing problems have a wide range of variety, Glock et al. [4] classified the variants of lot-sizing problems into many categories, of which, the following five categories are the most relevant.

1.2.1 Economic lot-sizing problem (ELSP). Single-item economic lot-sizing is the most well-known scheduling problem. The assumptions are that demand is deterministic for a finite planning horizon and backlogging is not allowed. The objective is to determine the number of units to order while minimizing the holding and ordering cost over the planning horizon [5].

1.2.2 Discrete lot-sizing problem (DLSP). The Discrete lot-sizing and scheduling problem has some fundamental assumptions that make it different from other lot-sizing problems. This type of problem is considered as a small bucket problem where one product can be produced at most in a given period. This can be referred to as all-or-nothing production policy where one product can be processed at full capacity or the facility process will be idle as shown in Figure 4. In each period only one product is being produced.

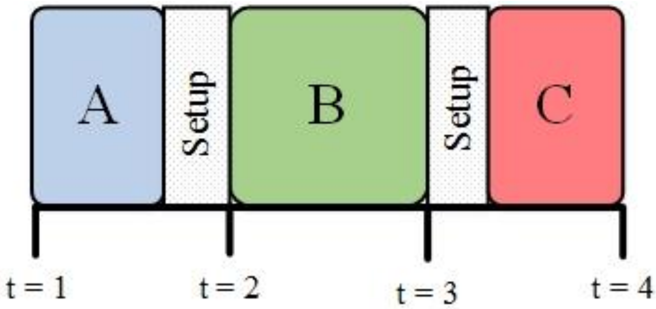

Figure 4 DLSP

Many products can be produced from a single production resource but with a certain capacity. The time horizon is finite and divided into discrete periods that are used for planning production. The demand for the products is deterministic but dynamic. The objective is to minimize the holding and changeover costs [6].

1.2.3 Capacitated lot-sizing and scheduling problem (CLSP). Capacitated lot-sizing problem (CLSP) is a large bucket problem where several products can be produced per time unit. The CLSP includes the sequence-dependent setup times and costs, in addition to the carry-over setup option [2]. Figure 5 provides an example of CLSP. It can be seen that many products can be produced in each period.

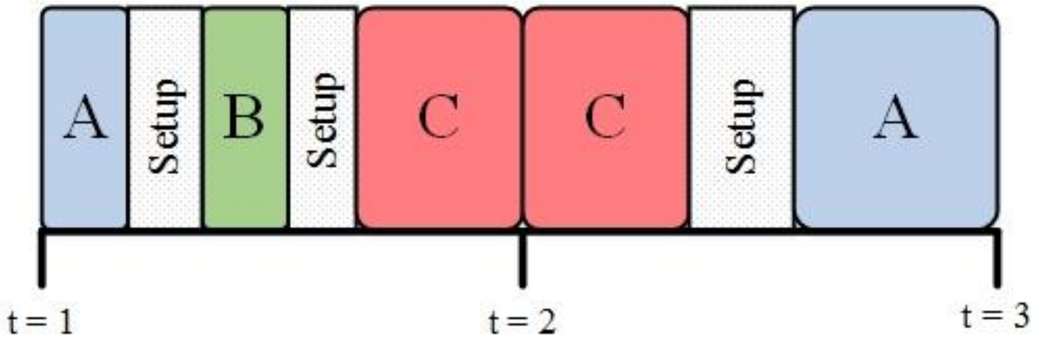

Figure 5 CLSP

1.2.4 Proportional lot-sizing problem (PLSP). This problem is a class of discrete model where multiple products can be produced to satisfy a dynamic demand. The proportional lot-sizing problem is considered as the most flexible small bucket model. It allows processing of two products only in the same period with a setup occurs when switching from the first to the second product. Figure 6 shows an example of PLSP. Sequence time and cost are important in this problem, therefore the model will minimize the setup and the holding cost [7]. From Figure 6, we can see that only two products are being produced at each period.

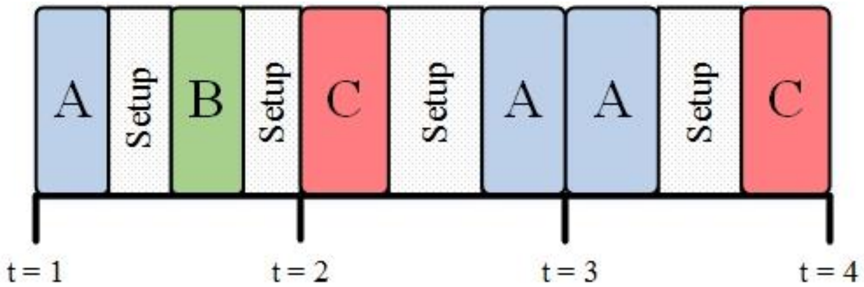

Figure 6 PLSP

1.2.5 General lot-sizing problem (GLSP). General lot-sizing problem (GLSP) is a hybrid between the DLSP and CLSP, where it is a mixed of a small bucket and big bucket problems. Each time unit in the planning horizon is divided into small micro periods with a variable length, where the sequence is obtained by assigning each lot to a micro period. The micro periods scheduling all together are considered as a macro period [8].

1.3 Solution Approaches

Bitran and Yanasse [9] studied the complexity of the capacitated lot sizing problem (CLSP). They found that when the size of the problem increases, the complexity of the model also increases. They addressed the computational complexity of the CLSP and proved that it is NP-Hard; meaning it is hard to solve a large practical problem within a reasonable computational time. Therefore, different approaches were developed to solve CLSP problems.

According to Buschkühl et al. [10], many approaches that are used to solve CLSP problems can be classified into five groups.

1.3.1 Mathematical programming heuristics (MP). MP heuristics are considered to be the most flexible among other approaches when it comes to a model extension. The algorithm will not stop until an optimal solution is found regardless of the computational time. Limiting the algorithm with a time frame is a simple method to develop in an MP-based heuristic. Limiting the time might generate infeasible solutions, however, these solutions can be considered as a better starting point to find a feasible solution. MP-Heuristics is divided into different approaches: Branch-and-Bound, Reformulations, Valid Inequalities, Fix-and-Relax, Rounding Heuristics, LP-based approaches and Dantzig-Wolfe and Column Generation [10].

1.3.2 Lagrangean relaxation heuristics. The Lagrangean Heuristic (LH) approaches are iterative solutions that apply Lagrangean relaxation (LR), in which the complicated constraints in the model are relaxed. Penalty cost is added to the objective function when these constraints are violated. Since it is an iterative approach, each iteration’s new lower bound is obtained based on the value of the lagrangean multipliers. A feasible solution is calculated and considered as the new upper bounds. Finally the multipliers are updated. Two approaches can be used for the LH: Lagrangean Relaxation Heuristics and Lagrangean Decomposition Heuristics [10].

1.3.3 Decomposition and aggregation heuristics. Decomposition and Aggregation approach work on the idea of dividing the problem into relatively small sub-problems, then the solutions of these problems are coordinated. Most of the decomposition approaches are time-based where it is combined with rolling the schedules. The idea is to divide the planning horizon into shorter periods. When a solution is found for the current period, the solution will be fixed and the problem will be solved again for the next periods. The Decomposition Heuristics can be time-based, item-based and resource-based, but the Aggregation Heuristics can only be Item-based and Resource-based. Bender's Decomposition is an example of these heuristics [10].

1.3.4 Metaheuristics. Metaheuristics can provide the flexibility and the ability to solve larger models. Metaheuristics use non-deterministic mechanisms that help to avoid trapping in limited areas of the solution space. The solution space may include infeasible solutions with constraint violations, but these violations are associated with penalty costs in the objective function. Metaheuristics use two main principles namely intensification and diversification, where the first improves the exploration of the search space, and the second allows for the exploitation of the accumulated search experience. Metaheuristics approaches include Local Search, Variable Neighborhood Search, Simulated Annealing Heuristics, Tabu Search Heuristics, Genetic Algorithms, Memetic Algorithms and Ant Colony Optimization Heuristics [10].

1.3.5 Problem-specific greedy heuristics. The greedy heuristics (GH) are spontaneous heuristics that start from scratch or from a given initial solution. GH follows a feasible routine and priority index to select the best candidate to move (increasing the lot size) while the solution stays in the feasible region. GH use two types of feasibility routines, namely, feedback mechanisms and look-ahead mechanisms. The first mechanism uses backward routines that force the infeasible production quantities to the earlier periods. Look-ahead mechanisms use forward routines that compute the minimum required inventory to avoid capacity violation in the future then schedule the production accordingly. Two approaches are used in GH, Constructive Greedy Heuristic and Improvement Greedy Heuristics [10].

1.4 Problem Statement

The Petrochemical Market is becoming competitive recently. In addition to that, the World Trade Organization increased the pressure and regulations on the petrochemical producers in the last few years due to the increment of the environmental laws. In order to survive and compete in the global market, companies started implementing different methods such as employing Enterprise-Resource-Planning (ERP) systems, supply-chain management and optimization tools such as Linear and

Integer Programming in order to increase the efficiency and effectiveness of their production systems [11].

In this research, integer linear programming is used to solve the problem that is concerned with multi-period planning and scheduling for a company that runs multiple affiliates to produce multiple grades of the same product. The company owns multiple warehouses and the raw material needed for production is provided by different suppliers. There is a setup cost for switching the production from one grade to another. This setup cost is sequence dependent.

When different grades are produced in a petrochemical facility, a non-prime material is formed during the setup changeover between the grades; and this material depends on the production sequence of the different grades. The work in this thesis will answer questions related to the amount of raw material to be purchased from each supplier, production plan for each affiliate where it specifies which grades to be produced, what is the lot size and in which sequence these grades should be produced, inventory levels of the different grades at each warehouse, and the selection of the warehouse that will satisfy each customer demand. A mathematical model will be developed and numerical example will be performed through an optimization software to validate and verify the model. The solution of the numerical example will be considered as a lower bound for the heuristic that will be developed to solve large scale problems.

1.4.1 Example of petrochemical company. Saudi Arabia Basic Industries Corporation (SABIC) is a manufacturing company that is located in Saudi Arabia; it is a chemical industrial company that manufactures polymers, fertilizers, and metals. SABIC owns several joint-ventures with other companies to produce certain grades of a product such as different grades of plastics. Figure 7 shows the supply chain stages for SABIC. When it comes to production planning, the Headquarter planning department is responsible for providing each affiliate with a monthly production plan. The plan depends on three factors: the expected inventory level of grades in the current period, production capacity in the next period and the expected demand in the next period. The department starts matching the supply with the demand for each affiliate after taking into consideration the aforementioned factors. When the production plan is

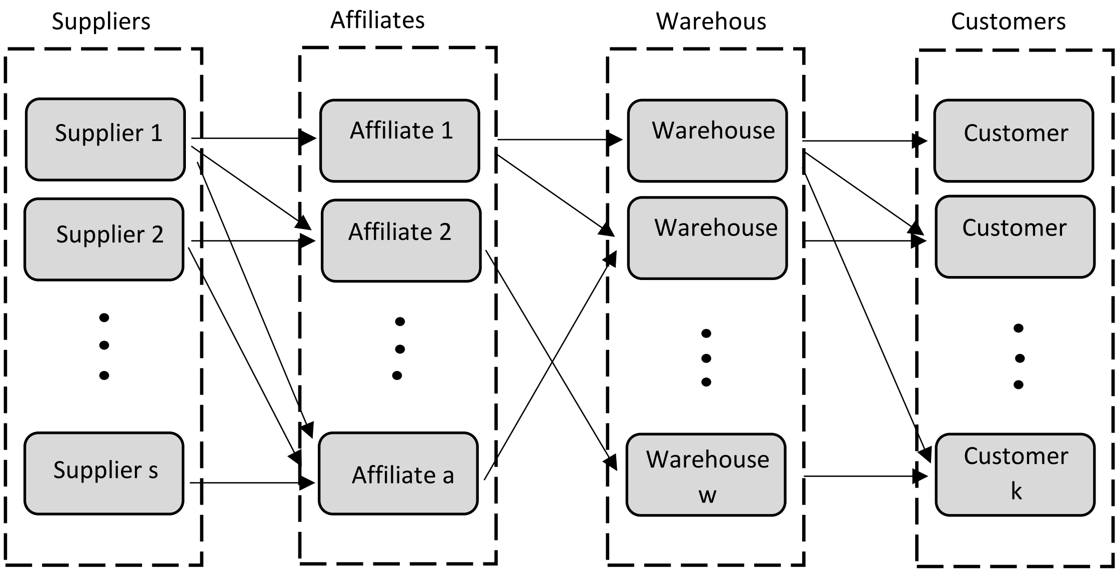

Figure 7 SABIC Supply Chain

ready, the business department modifies the plan by a certain percentage to consider the uncertainty in the market. After the plan is finalized, each affiliate is requested to produce a certain amount of the different grades. The disaggregation of the plan is made by the production engineers in each affiliate.

Grades are produced through lots where the size and the schedule of the lots are planned in a way that minimizes the transition cost. The transition cost is a setup cost that occurs when different grades are produced. This cost is considered expensive due to the nature of the process. The traditional process of producing petrochemical products is comprised at three stages as shown in Figure 8.

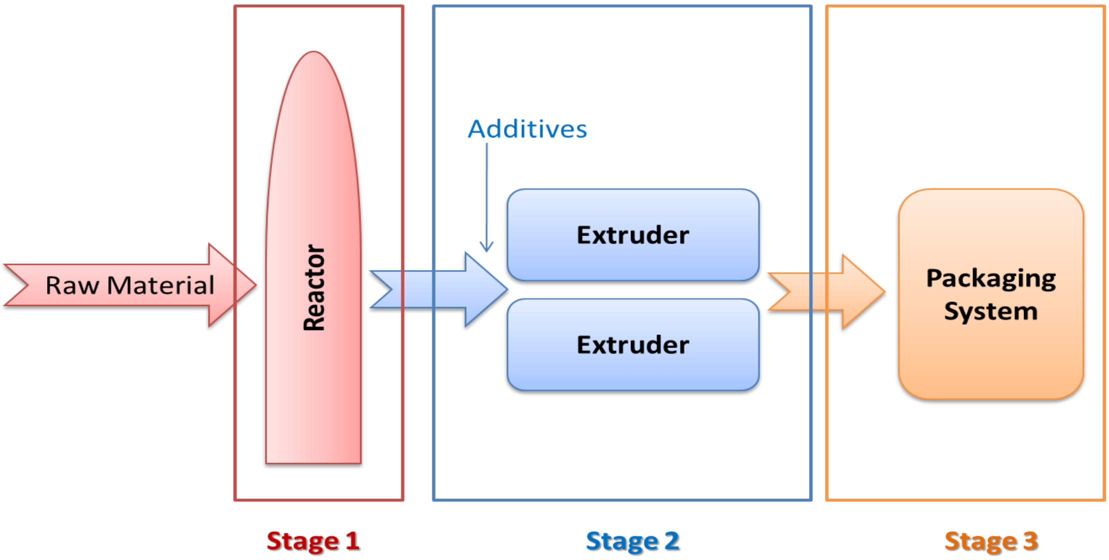

Figure 8 process map at SABIC affiliates

In stage one, the raw materials arrive and enter a reactor of a known capacity to be processed. Many conditions such as temperature, pressure, and feed rates of raw materials must be calibrated inside the reactor in order to produce a certain grade. Changing these conditions can result in a much different type of grades, where these grades may differ in physical and chemical properties. Producing different grades needs a changeover period in the reactor and in this period an amount of transitional material is produced.

The quantity of this material depends on the sequence of the production, i.e. if a transition is made from grade one to grade two, the quantity of the material produced in the transition process is lower than the one produced when a transition is made from grade one to grade three or higher. Therefore, this problem is considered as sequence-dependent setup quantity, because the transitional material quantity that occurs between the change-overs depends on the sequence of the production.

In Stage two, some additives are added to the grades. These additives can change the specifications of the same grade to transform one grade into sub-grades. Finally, in stage three, the output of stage two will be packed into different packaging types. This research will focus on stage one only because the major production cost and the sequence-dependent setups occurs in this stage.

1.5 Research Objective and Significance

The objective of the research is to develop a Mixed-Integer Linear Program (MILP) that integrates the planning process of supplier selection, lot-sizing and sequencing decisions and inventory levels in addition to the logistics functions of transportation and warehousing. This model will optimize the production planning and scheduling decisions in order to maximize the net profit among the whole supply-chain.

In the objective function, the net profit will be maximized by subtracting the total revenue generated by the different affiliates from the costs of raw materials, setup costs, inventory holding costs and transportation costs over the planning horizon. The model will be solved using an optimization software.

Bitran and Yanasse [9] proved that this problem is NP-Hard, therefore a heuristic will be developed to obtain a feasible solution with a good quality in a reasonable computational time. Then a sensitivity analysis will be performed on different problems to study the effects of the model parameters on the planning decisions and the performance of the developed heuristic.

The main contributions of this research include the following:

1. Enhance the literature of the capacitated lot-sizing and scheduling problem with a new model that integrates the decisions of scheduling and planning of lot-sizing with the decisions of the logistics functions such as transportation and warehousing. This integration will minimize the overall supply-chain cost.
2. Develop a Mixed-Integer Linear program that answers questions related to supplier selection for raw material, production plan for each plant, inventory levels of the different grades at each warehouse, and warehouse selection to satisfy demand.
3. Develop a Heuristic that will help to obtain a good quality solution for large problem instance within a reasonable computational time.
4. The developed model will have a great research potential since it will be considered as a general model that combines the operational (lot-sizing planning and scheduling) and tactical (supplier selection and transportation routes) decisions.

1.6 Methodology

The following steps will be followed to achieve the outcomes of this research:

Step 1: Review the literature related to lot-sizing problems, capacitated lot-sizing and scheduling problems and lot-sizing solution approaches.

Step 2: Formulate a mixed-integer linear programming model by defining the model assumptions, parameters, decision variables, objective function, and constraints.

Step 3: Code the formulated model using Lingo optimization software and testing it’s functionality by solving a numerical example.

Step 4: Develop a solution approach to solve large instance problems with good quality solution in a reasonable computational time.

Step 5: Perform sensitivity analysis to test the effect of varying the model’s inputs on the total supply-chain cost and the developed heuristic performance.

1.7 Thesis Organization

In Chapter 1, an introduction about lot sizing problems is presented where the different characteristics, classes and solution approaches are discussed. In addition to the research problem, the objective, the significance and the methodology are stated. Chapter 2 is dedicated to surveying all relevant literature about capacitated lot sizing problems with the solution approaches. Chapter 3 introduces the developed mathematical model that represent the problem mentioned in this research with an illustrative numerical example. It also presents the solution method that was developed to solve the mathematical model in a reasonable time. Chapter 4 represents the sensitivity analysis of the model and the heuristic. Finally Chapter 5 contains the conclusion and suggested future work.

Chapter 2: Literature Review

The capacitated lot sizing problem can have different configurations with different levels of complexity. The literature reviewed in this work will focus on the single level, dynamic demand, capacitated lot sizing with different machine configurations, setup cost and time dependence and the solution algorithm.

This chapter is divided into five sections. The first three sections are the different machine configurations, the fourth is about distinctive properties in the capacitated lot sizing problem, and the fifth section is about the application of capacitated lot sizing problem in the petrochemical industry. Table 2 summarizes the relevant papers that were reviewed.

Table 2 Summary of Relevant papers in the literature

Author	Level		Plants		Machines			Setup Cost		Setup Time			Solution*
	single	**multiple**	**single**	**multiple**	**single**	**Parallel**	**Flow Shop**	**Dependent**	**Independent**	**Dependent**	**Independent**	**Carry-over**	
Almada -Lobo **[12]**	x		x		x	x		x		x			H
Kwak and Jeong **[13]**	x		x		x			x		x			HA
Mirabi **[14]**	x		x		x			x					SA
Shim **[15]**	x		x		x			x		x		x	H

* H: Heuristics, HA: Hierarchical Approach, SA: Simulated Annealing, MH: Metaheuristics
GA: Genetic Algorithms, AC: Ant Colony, DP: Dynamic Programming, FO: Fix and Optimize

Table 2 Summary of Relevant papers in the literature (Continued)

Author	Level		Plants		Machines			Setup Cost		Setup Time			Solution*
	single	**multiple**	**single**	**multiple**	**single**	**Parallel**	**Flow Shop**	**Dependent**	**Independent**	**Dependent**	**Independent**	**Carry-over**	
Almada-Lobo and James **[2]**	x		x		x			x		x			MH
Sung & Maravelias **[16]**	x		x		x			x		x		x	
Süer et al. **[17]**	x		x		x			x		x			GA
Almada-lobo et al. **[18]**	x		x		x			x		x		x	H
Almeder **[19]**		x	x		x			x		x			AC
Sambasivan **[20]**	x		x		x				x		x		H
Xiao et al. **[21]**	x		x			x		x		x			SA
Fiorotto & Araujo **[22]**	x		x			x			x		x		H
Quadt & Kuhn **[23]**		x	x			x		x		x		x	

* H: Heuristics, HA: Hierarchical Approach, SA: Simulated Annealing, MH: Metaheuristics
GA: Genetic Algorithms, AC: Ant Colony, DP: Dynamic Programming, FO: Fix and Optimize

Table 2 Summary of Relevant papers in the literature (Continued)

Author	Level		Plants		Machines			Setup Cost		Setup Time			Solution*
	single	**multiple**	**single**	**multiple**	**single**	**Parallel**	**Flow Shop**	**Dependent**	**Independent**	**Dependent**	**Independent**	**Carry-over**	
Toledo & Armentano **[24]**	x		x			x			x		x		H
Mateus **[25]**	x		x			x		x		x			H
Marinelli **[26]**	x		x			x			x		x		H
Mohammadi et al. **[27]**	x		x				x	x		x			H
Babaei et al. **[28]**	x		x				x	x		x		x	GA
Ramezanian & Mehrabad **[29]**	x		x				x	x		x		x	H
Mohammadi & Ghomi **[30]**	x		x				x	x		x			H
Mohammadi et al. **[31]**	x		x				x	x		x			H

* H: Heuristics, HA: Hierarchical Approach, SA: Simulated Annealing, MH: Metaheuristics
GA: Genetic Algorithms, AC: Ant Colony, DP: Dynamic Programming, FO: Fix and Optimize

Table 2 Summary of Relevant papers in the literature (Continued)

Author	Level		Plants		Machines			Setup Cost		Setup Time			Solution*
	single	**multiple**	**single**	**multiple**	**single**	**Parallel**	**Flow Shop**	**Dependent**	**Independent**	**Dependent**	**Independent**	**Carry-over**	
Almada-Lobo et al. **[32]**	x		x		Multiple			x		x		x	H
Nascimento et al. **[33]**	x			x		x							H
Zhang **[34]**	x		x		x								DP
Almedera et al **[35]**		x	x		x			x		x		x	
Tempelmeier & Hilger **[36]**	x		x		x							x	H
Kantas et al. **[37]**	x		x		x								
Lu et al. **[38]**	x		x		x				x				
Xiao et al. **[39]**	x		x			x		x		x		x	FO
Pan et al. **[40]**	x		x		x								
Akbalik **[41]**	x		x		x								

* H: Heuristics, HA: Hierarchical Approach, SA: Simulated Annealing, MH: Metaheuristics
GA: Genetic Algorithms, AC: Ant Colony, DP: Dynamic Programming, FO: Fix and Optimize

Table 2 Summary of Relevant papers in the literature (Continued)

Author	Level		Plants		Machines			Setup Cost		Setup Time			Solution*
	single	**multiple**	**single**	**multiple**	**single**	**Parallel**	**Flow Shop**	**Dependent**	**Independent**	**Dependent**	**Independent**	**Carry-over**	
Haugen et al. **[42]**	x		x		x								
Al Fares **[43]**	x		x		x			x		x			
Al Fares **[11]**	x		x		x			x		x			
Alqahtani et al. **[44]**	x		x		Multiple			x		x			
Gupta & Magnusson **[45]**	x		x		x			x		x			H
Almada-Lobo et al. **[46]**	x		x		x			x		x			
Thesis Objective	**x**			**x**	**x**			**x**		**Quantity Dependent**		**x**	**H**

* H: Heuristics, HA: Hierarchical Approach, SA: Simulated Annealing, MH: Metaheuristics
GA: Genetic Algorithms, AC: Ant Colony, DP: Dynamic Programming, FO: Fix and Optimize

2.1 CLSP with Single Machine Configuration

James and Almada-Lobo [12] proposed two models that differ in the machine configuration. The first consider a single machine capacitated lot-sizing with sequence dependent time and cost, and the second consider a parallel machine capacitated lot-sizing with sequence dependent time and cost. Both of the models are considered NP-Hard. Therefore, a new iterative MIP-based neighborhood search heuristics was developed to tackle the problem under study. Kwak and Jeong [13] considered a model that deals with a single machine capacitated lot-sizing with sequence dependent time and cost. A special form of setup cost is considered where the sequence-dependent setup time increases with the size of the product. A two level hierarchical approach is used to solve the model.

Mirabi [14] modeled a single machine capacitated lot-sizing and scheduling problem with sequence-dependent setup costs. They used a hybrid simulated annealing to solve the Hard-NP. Shim [15] considered a single machine capacitated lot-sizing and scheduling problem with sequence-dependent cost and time, in addition to the carry-over setup. Then a two-stage heuristic is suggested to solve the model. Almada-Lobo and James [2] addressed a problem where commercial solvers failed to solve Large-sized NP-Hard problems. The multi-product capacitated lot sizing and scheduling problem with single machine configuration and sequence-dependent setup times and costs were solved using a tabu search and a variable neighborhood search meta-heuristic. A set of computational experiments were performed to show the effectiveness of the approach.

Sung and Maravelias [16] developed a mathematical model that considered a single machine configuration with sequence-dependent setup costs, carryover option and backlogging. Moreover, setups can start in one period and finish in the next period or take longer than one period. Süer et al. [17] implemented multi-chromosome crossover strategy to solve CLSP in short computational time. The model considered a single machine with independent sequence setup time, cost, including labor cost and ordering cost. Their crossover strategy was proven to perform well in large instance problems. Almada-lobo et al. [18] proposed a five step heuristics to find a feasible solution and generates a high quality solution for a single machine CLSP. The model considered a sequence-dependent setup time and cost in addition to the carry over setup.

Almeder [19] presented a new approach in ant algorithms. The approach is faster and more powerful in compared to other solution approaches, Max-Min ant system was developed to solve a Multi-level Capacitated Lot Sizing Problem that has a single machine configuration. The setup cost and time is sequence independent, and the model is able to handle overtime if needed. Sambasivan [20] addressed a real world problem in a steel manufacturing company, where a formulation for a multiple plant CLSP with a single machine configuration is developed. The setup time and cost is independent. The model has the flexibility of inter-plant transfers where plants can transfer quantities from/to other plants. Then, a Lagrangean-based heuristic was developed to solve the NP-hard problem.

2.2 CLSP with Parallel Machines Configuration

Recently Xiao et al. [21] examined the parallel machine configuration with sequence-dependent setup times, time windows, machine eligibility and preference in addition to the backlogging option. These properties are common in the semiconductor manufacturing industry. The model is solved using the hybrid Lagrangean-simulated annealing-based heuristic algorithm.

Fiorotto and Araujo [22] considered a capacitated lot sizing problem with a single level manufacturing, with multiple products. Independent setup time that was not affected by the sequence of the lot sizes and a configuration of parallel machine. Then the model was solved using a Lagrangean Heuristics with a reformation of the model as the shortest path problem. The results of the Heuristics were compared with high-performance Mixed Integer Program software.

Quadt and Kuhn [23] presented a CLSP formulation for multi-level products that were processed on parallel machines configuration. The model considered sequence-dependent with carry-over setup and backlogging. Also it points out the use of overtime in CLSP. Toledo and Armentano [24] developed a Lagrangean relaxation of the capacity constraints and subgradient optimization to solve a CLSP with parallel machines configuration. The setup cost and time were sequence independent.

The heuristics showed that a high number of feasible solutions with high quality can be found in a short time. Mateus [25] considered a parallel machine's configuration for CLSP with sequence dependent setup cost and time. GRASP heuristic was proposed to solve the NP-hard problem. Marinelli [26] developed a two stage heuristics optimization to solve CLSP with parallel machines and buffer stock. The setup time and cost were sequence independent, the model was applied in a packaging company for yogurt. The results of the heuristics were very effective and showed good performance.

2.3 CLSP with Flow Shop Configuration

Mohammadi et al. [27] presented a capacitated lot-sizing model with a sequence-dependent cost and time in a pure flow shop configuration. Solving the CLSP is equivalent to solving the multiple dependent Traveling Salesman Problems (TSP). Mohammadi et al. [27] proved that since TSP is NP-Hard, CLSP is NP-Hard as well. Five MIP-based Heuristics were proposed, classified into two categories. The first category dealt with large instances of problems while the second category considered lot-sizing problems with a flow shop configuration. Babaei et al. [28] proposed a model that considered a flow shop configuration with sequence-dependent costs, setup carry-over and backlogging. This model was solved using a genetic algorithm.

Ramezanian and Mehrabad [29] developed a multi-period capacitated lot-sizing in a flow shop configuration. The novelty of the work was in proposing an efficient mathematical model that reduced the solution complexity. Sequence-dependent setups in terms of time and cost were introduced in the model, in addition to the carry-over option. The model was solved using Heuristic algorithms based on rolling horizon and the performance was evaluated through the trial of the variant scale of the problem.

Mohammadi and Ghomi [30] considered a capacitated lot-sizing and scheduling problem in a flow shop configuration with sequence-dependent setups. The contribution was in developing a genetic algorithm-based heuristic that solved NP-Hard problem. The algorithms combined genetic algorithms with rolling horizon approach. The evaluation of the effectiveness of the method was by comparing the final results of this method with others heuristics. The results were superior in comparison to the others.

Mohammadi et al. [31] proposed a new algorithmic approach to solve a capacitated lot-sizing and scheduling problem with sequence-dependent setups in a flow shop configuration. The approach was held using two solution algorithms, the rolling horizon approach and a specific heuristics.

2.4 CLSP with Distinctive Properties

Almada-Lobo et al. [32] addressed a problem that occurs in the glass container industry where the furnace used is not allowed to be idle. The model considered a multiple machine configuration with a sequence-dependent cost and time in addition to a carry-over setup. The raw materials that were used in the manufacturing process were similar, in other words, all products shared the same resource. Since CLSP is considered NP-Hard, the solution method used was a Lagrangean decomposition based heuristic. Nascimento et al. [33] considered a problem where multi plants produce multi products with the possibility of transfers between the plants, in other words, a plant can produce and send to plant B. Capacitated lot-sizing and scheduling were considered, however with sequence independent setups. Greedy Randomized Adaptive Search Procedure heuristic, as well as a Path-Relinking Intensification procedure, were developed to solve this NP-Hard problem. In addition, this heuristic was applied on parallel machine configuration.

Zhang [34] studied a capacitated lot sizing problem with an option of outsourcing. Production capacity was considered as constant and the outsourcing capacity was considered uncapacitated. A dynamic programming-based algorithm was developed to solve the model while considering a polynomial time. Almedera et al [35] recognized an important dimension, lead-time that was not often considered in classical capacitated lot-sizing and scheduling problems. The multi-level CLSP models usually do not mimic the correct resource requirements or the precedence relations. Therefore two models were presented; one that considered batch production with lead times and the other considered the allowance of lot-streaming. Results were compared with current models and cost savings of 30–40 % were observed.

Tempelmeier and Hilger [36] considered a situation where lot-sizing was required but with a present of a stochastic dynamic demand. Multi-items with constant capacity and backorders were considered. The model was developed using linear programming, and solved using a fix-and-optimize heuristics. Kantas et al. [37] formulated a model for capacitated lot-sizing that took into consideration CO2 emissions and water excessive. The model minimized the cost of producing ethanol production while taking into consideration the different sources of biomass that are considered as the raw materials while considering the different impact of each source on the CO_2 emissions level.

Lu et al. [38] integrated the classical capacitated lot-sizing model with preventive maintenance consideration. Many plants require the system reliability to be above a certain value as this can impact the production in many ways. The reliability constraints were linearized to have a mixed-integer linear program. Three stage heuristics were developed to solve the model including a Lagrangian-based heuristic to solve the CLSP. Numerical experiments were performed to support the efficiency of the heuristics. Xiao et al. [39] examined a capacitated lot-sizing and scheduling problem with sequence-dependent setup times in a parallel machine configuration. The contribution was in adding the machine eligibility and preference constraints. These constraints frequently happen in the semiconductor manufacturing industry. A mixed integer program was developed and solved by fix-and-optimize algorithms to get the solutions.

Pan et al. [40] developed a capacitated lot sizing model with closed loop supply chain. Customers can return the products to the manufacturer, where they would be remanufactured or disposed. Therefore customers demand can be satisfied from new products or remanufactured products. Akbalik [41] studied a special case of capacitated lot sizing problems where a single item can be processed on different machines. Three models were developed to mimic this case. Haugen et al. [42] developed a model that considered pricing in capacitated lot sizing problems, where the price of the final product was considered in the optimization problem to maximize the total profit.

2.5 CLSP Applied in Petrochemical Industry

Al Fares [43] developed a mathematical model that optimized the planning of multiple petrochemical plants that produced multiple grades. Transitional material depended on the sequence of the production plan in the model.

The model considered the demand, the capacity, the raw material availability, and the sequence constraints in order to optimize the mix of grades to be produced in each plant. The quantity of each grade to be produced, and the sequence of these grades. However, the model did not consider a multiple period planning and did not link it to the inventory control. Another model was presented by Al Fares [11] where the option of purchasing different raw materials from different suppliers was considered. The model optimized the grade selection in each plant, the quantity to be produced, and the sequence of the production. The model was applied to a real-life data from a petrochemical company. Even though the paper did not encounter the shortfalls of the previous paper, since the model did not consider the multiple planning periods and the inventory linkage.

Alqahtani et al. [44] considered a Mixed Integer Linear Program model that optimized the planning and scheduling of a continuous production process in petrochemical factories called a polyethylene process. The model considered multiple reactors that could produce multiple polyethylene grades. Furthermore, it focused on reducing the overall losses of the underutilized capacities and the potential risk that could occur during the production or the maintenance period. Due to the focus strategy, the model optimized the strategy of production planning over a long period of time. The model was applied to a case study that consisted of six reactors, twelve grades and a period of one year. The overall utilization of the plant was improved because of the new planning strategy that was proposed by the model.

Gupta and Magnusson [45] considered that CLSP as NP-hard, therefore in addition to developing a mathematical model that considered a single machine with multiple products over multiple periods, a heuristics were developed to solve a large instance problems. The heuristics depended on three stages, Initialize, sequence and improve.

The performance of the algorithm was improved whenever the size of the problem increases, however, it was noticed by Almada-Lobo et al. [46] that the model did not avoid disconnected subtours, therefore new constraints were introduced to eliminate the disconnected subtours.

2.6 Chapter Summary

After reviewing the literature, it was concluded to the best of our knowledge that over forty-six papers mostly considered the capacitated lot-sizing and scheduling problem with sequence-dependent setup costs that have a single plant and without the connection to the logistics function (suppliers and the warehouses). Therefore a Mixed-Integer Linear program will be developed to fill the gap of having multiple suppliers, multiple affiliates, multiple warehouses and multiple customers with the sequence-dependent setup costs. The model will deal with supplier selection for raw material, production plan for each plant, inventory levels of the different grades at each plant, and warehouse selection to satisfy demand.

Chapter 3: Mathematical Model

The objective of this chapter is to develop and solve a mathematical model that will answer questions related to supplier selection for raw material, production plan for each plant, inventory levels of the different product grades at each warehouse, and warehouse selection to satisfy demand. Accordingly, the solution of this model will optimize the net profit by subtracting the total revenue from the following costs:

- The total transportation cost from the suppliers to the plants, from the plants to the warehouses and from the warehouses to the customers.
- The cost of raw materials purchased from each supplier.
- The cost of setup in each plant.
- The holding cost of inventory in each warehouse.

Model assumptions, indices, parameters and decisions variables will be presented next before the introduction of the model.

3.1 Model Assumptions

The following assumptions are considered in the development of the mathematical model.

1. All the parameters used in the model are deterministic.
2. Lot sizing will be for multiple grades.
3. Planning horizon is finite with T periods.
4. Each period length is defined by the corresponding total available capacity.
5. Setup cannot start in one period and end in the next period.
6. The manufacturing level is considered as a single level, where the products need one operation to be transformed from raw materials into finished goods.
7. Production can start from any grade in each period.
8. Each product can have only one setup at each period.
9. Setup carry-over is allowed.
10. Lead times are not considered.
11. Backlogging and overtime are not allowed.
12. Each supplier has a limited capacity.
13. Several affiliates are available for production.
14. Each affiliate has a limited capacity.

15. Each affiliate can produce all the grades.
16. Each grade has a different profit per unit.
17. Each grade has different demand.
18. Each grade has a different usage of raw material.
19. All grades need to be transported to the warehouse then to the customer.
20. Inventory is allowed in the warehouses only.

3.2 Problem Parameters and Indices

The following are the indices that are used in the mathematical model:

$\boldsymbol{s}$	$Supplier,$	$s = 1, \dots, S$
$\boldsymbol{r}$	$Raw\ Material,$	$r = 1, \dots, R$
$\boldsymbol{a}$	$Affiliate,$	$a = 1, \dots, A$
$\boldsymbol{g, h}$	$Grade,$	$g = 1, \dots, G$
$\boldsymbol{w}$	$Warehouse,$	$w = 1, \dots, W$
$\boldsymbol{k}$	$Customer,$	$k = 1, \dots, K$
$\boldsymbol{t}$	$Period,$	$t = 1, \dots, T$

The following are the parameters that are used in the mathematical model:

$\boldsymbol{C_{sar}^{trans}}$	Transportation cost per ton of raw material *r* from supplier *s* to affiliate *a*.
$\boldsymbol{C_{awg}^{trans\prime}}$	Transportation cost per ton of grade *g* from affiliate *a* to warehouse *w*.
$\boldsymbol{C_{wkg}^{trans\prime\prime}}$	Transportation cost per ton of grade *g* from warehouse *w* to customer *k*.
$\boldsymbol{C_{sar}^{raw}}$	Purchasing Cost per ton of raw material *r* from supplier *s* to affiliate *a*.
$\boldsymbol{P_{agt}^{prime}}$	Price per ton of prime grade *g* in affiliate *a* during time period *t*.
$\boldsymbol{C_{aght}^{nprime}}$	Production cost per ton of none-prime grade *g* that is formed in transition between grade *g* and h in affiliate *a* during time period *t*.
$\boldsymbol{C_{wgt}^{inv}}$	Inventory carrying cost per ton of grade *g* in warehouse *w* during time period *t*.
$\boldsymbol{\lambda_{sr}}$	Availability of raw material *r* in tons from supplier *s* per period which is constant across all periods.

$\boldsymbol{\lambda'_{at}}$	Production capacity of affiliate *a* during time period *t*.
$\boldsymbol{O_{aght}}$	Amount of none-prime grade *g* in tons that is formed in transition between grade *g* and *h* in affiliate *a* during time period *t*.
$\boldsymbol{\beta_{arg}}$	Quantity needed of raw material *r* to produce one ton of grade *g* in affiliate *a*.
$\boldsymbol{D_{gkt}}$	Demand of grade *g* for customer *k* during time period *t*.
$\boldsymbol{EX_t}$	Excess production capacity of all affiliates in period *t*.
$\boldsymbol{\rho_{gt}}$	Percentage of the excess capacity allocated to grade *g* at each period *t*.

3.3 Problem Decision Variables

The following are the decision variables of the mathematical model:

$\boldsymbol{F_{sart}}$	Tons of raw material *r* shipped from supplier *s* to affiliate *a* during time period *t*.
$\boldsymbol{F'_{awgt}}$	Tons of grade *g* shipped from affiliate *a* to warehouse *w* during time period *t*.
$\boldsymbol{F''_{wkgt}}$	Tons of grade *g* shipped from warehouse *w* to customer *k* during time period *t*.
$\boldsymbol{X_{agt}}$	Tons of grade *g* produced in affiliate *a* during time period *t*.
$\boldsymbol{Z_{aght}}$	A binary variable that equals 1, if transition is made between grade *g* and *h* in affiliate *a* during time period *t*, 0 otherwise.
$\boldsymbol{\alpha_{agt}}$	A binary variable that equals 1, if grade *g* is produced first in affiliate *a* during time period *t*, 0 otherwise.
$\boldsymbol{V_{agt}}$	Auxiliary variable of grade *g* in affiliate *a* during time period *t* that is used to validate the constraint of sub-tour elimination.
$\boldsymbol{I_{wgt}}$	Inventory level of grade *g* in warehouse *w* during time period *t*.

3.4 Model Formulation

- ***Objective Function***

The objective function (1) is divided into five parts, the first part considers the profit generated from each grade in each affiliate at each period, the second part considers the transportation costs from the suppliers to the affiliates, the affiliates to the warehouses and the warehouses to the customers. The third part considers the cost of the raw materials that can be purchased from different suppliers. The fourth part considers the transition cost that occurs when different grades are produced. The last part considers the inventory cost in each warehouse.

$$Max = \left[\sum_{a=1}^{A}\sum_{g=1}^{G}\sum_{t=1}^{T} P_{agt}^{prime} * X_{agt}\right] - \left[\sum_{s=1}^{S}\sum_{a=1}^{A}\sum_{r=1}^{R}\sum_{t=1}^{T} C_{sar}^{trans} * F_{sart} + \sum_{a=1}^{A}\sum_{w=1}^{W}\sum_{g=1}^{G}\sum_{t=1}^{T} C_{awg}^{trans'} * F'_{awgt} + \sum_{w=1}^{W}\sum_{k=1}^{K}\sum_{g=1}^{G}\sum_{t=1}^{T} C_{wkg}^{trans''} * F''_{wkgt}\right] - \left[\sum_{s=1}^{S}\sum_{a=1}^{A}\sum_{r=1}^{R}\sum_{g=1}^{G} C_{sar}^{raw} * F_{sart}\right] - \left[\sum_{a=1}^{A}\sum_{g=1}^{G}\sum_{h=1, h\neq g}^{G}\sum_{t=1}^{T} C_{aght}^{nprime} * O_{aght} * Z_{aght}\right] - \left[\sum_{w=1}^{W}\sum_{g=1}^{G}\sum_{t=1}^{T} C_{wgt}^{inv} * I_{wgt}\right] \qquad (1)$$

- ***Supplier Constraints***

$$\sum_{g=1}^{G} \beta_{arg} * X_{agt} = \sum_{s=1}^{S} F_{sart} \qquad \forall\, a, r, t \qquad (2)$$

$$\sum_{a=1}^{A} F_{sart} \leq \lambda_{sr} \qquad \forall\, s, r, t \qquad (3)$$

Constraints (2) and (3) ensure that the required amount of each raw material *r* in each affiliate *a* to produce each grade *g* will be transported from the different suppliers *s* at each period *t* without exceeding the supplier’s capacity.

- ***Affiliate Constraints***

$$\sum_{g=1}^{G} X_{agt} + \sum_{g=1}^{G} \sum_{h=1,h\neq g}^{G} O_{aght} * Z_{aght} \leq \lambda'_{at} \qquad \forall\, a\,, t \qquad (4)$$

The total amount produced at each period *t* of prime grades and none-prime grades in each affiliate *a* cannot exceed its capacity.

$$X_{agt} \leq \lambda'_{at} * \left(\sum_{h=1,h\neq g}^{HG} Z_{ahgt} + \alpha_{agt} \right) \qquad \forall\, a\,, g\,, t \qquad (5)$$

$$\sum_{h=1,h\neq g}^{G} Z_{ahgt} + \alpha_{agt} \leq 1 \qquad \forall\, a\,, g\,, t \qquad (6)$$

$$\sum_{g=1}^{G} \alpha_{agt} = 1 \qquad \forall\, a\,, t \qquad (7)$$

$$Z_{aght} \leq X_{aht} \qquad \forall\, a\,, g, h\,, t \qquad (8)$$

$$\alpha_{agt} + \sum_{h=1,h\neq g}^{G} Z_{ahgt} = \alpha_{agt+1} + \sum_{h=1,h\neq g}^{G} Z_{aght} \qquad \forall\, a\,, g\,, t \qquad (9)$$

$$V_{agt} + (N * z_{aght}) - (N-1) - (N * \alpha_{agt}) \leq V_{aht} \quad \forall\, a\,, g\,, h \neq g, t \quad (10)$$

Constraint (5) guarantees that grade *g* in each affiliate *a* can be produced at period *t* only if the machine is setup first for the grade or at least one setup for grade *g* exists. Constraint (6) ensures that each grade *g* in each affiliate *a* at period *t* can have one setup either from the first setup or by a transition from another grade. Constraint (7) ensures that the setup at the beginning of each period *t* can be done only for one grade *g* in each affiliate *a*. Constraint (8) ensures that there is no transition to any grade unless this grade is produced. Constraint (9) represents the carry-over setup. This constraint will be explained through an example. Assuming the following scenario:

- One affiliate.
- Three grades.
- Two periods.

- The demand for grade 1 and 2 in period 1 is greater than zero.
- The demand for grade 2 and 3 in period 2 is greater than zero.
- For Grade 1 period 1: the model will start with producing grade 1 α_{111} = 1 and force a changeover to grade 2 z_{1121}= 1.
- For Grade 2 period 1: a transition has happened from grade 1 z_{1121}= 1 and no other transitions will be done, which means the setup will be carried over to period 2 α_{122} = 1.
- For Grade 2 period 2: the model will start with producing grade 2 α_{122} = 1 and force a changeover to grade 3 z_{1232}= 1.
- For Grade 3 period 2: a transition occurred from grade 2 z_{1232}= 1 and no other transitions will be done, which mean the setup will be carried over to period 3 α_{133} = 1.

In general, if there is no transition from one grade to another (i.e. $\sum_{h=1,h\neq g}^{H} z_{ahgt} = \sum_{h=1,h\neq g}^{H} z_{aght} = 0$), the state of the machine is carried over to the next period (i.e. $\alpha_{agt} = \alpha_{agt+1} = 1$), if there is a transition to a grade g only (i.e. $\sum_{h=1,h\neq g}^{H} z_{ahgt} = 1$), the setup of grade g will be considered the first setup in the next period (i.e. $\alpha_{agt+1} = 1$), If there is a transition to a grade h (i.e. $\sum_{h=1,h\neq g}^{H} z_{aght} = 1$) and no transition to a grade g (i.e. $\sum_{h=1,h\neq g}^{H} z_{ahgt} = 0$), this means that this setup is the first in period t (i.e. $\alpha_{agt} = 1$). Constraint (10) ensures that the model avoid the disconnected sub-tours and sustain the feasibility. This constraint is known for the Traveling Salesman Problem and was proposed by Nemhauser and Wolsey [47]. This constraint ensures that each grade can be produced only once each period in each affiliate.

- ***Excess Capacity Constraint***

$$EX_t = \sum_{a=1}^{A} \lambda'_{at} - \sum_{a=1}^{A}\sum_{g=1}^{G} X_{agt} - \sum_{a=1}^{A}\sum_{g=1}^{G}\sum_{h=1,h\neq g}^{G} O_{aght} * Z_{aght} \quad \forall\ t \qquad (11)$$

Constraint (11) calculates the total excess capacity in all affiliates after satisfying the total demand at each period t.

- ***Warehouse Constraints***

$$X_{agt} = \sum_{w=1}^{W} F'_{awgt} \qquad \forall\, a\,, g\,, t \qquad (12)$$

$$I_{wgt-1} + \sum_{a=1}^{A} F'_{\boldsymbol{awgt}} = I_{wgt} + \sum_{k=1}^{K} F''_{wkgt} \qquad \forall\, w\,, g\,, t \qquad (13)$$

Constraint (12) ensures that all the produced grades *g* in each affiliate *a* at each period *t* should be transported to the warehouses *w*. Constraint (13) represents the inventory balance for each grade *g* in each warehouse *w* at each period *t*.

- ***Customer Constraints***

$$\sum_{w=1}^{W} F''_{wkgt} \geq D_{gkt} \qquad \forall\, g\,, k\,, t \qquad (14)$$

$$\sum_{w=1}^{W} \sum_{k=1}^{K} F''_{wkgt} = \sum_{k=1}^{K} D_{gkt} + \rho_{gt} * EX_t \qquad \forall\, g\,, t \qquad (15)$$

Constraint (14) indicates that the total amount of grade *g* transported from all warehouses should be at least equal to the demand of each customer *k* at each period *t*.

Constraint (15) Make sure that the full capacity will be utilized if an excess capacity exists and it is allocated to certain grades with certain percentages.

3.5 Numerical Example

In this section, an illustrative example is presented and tested to present the developed mathematical model outcomes. The size of the parameters of this example is as follow.

- Number of Suppliers = 4
- Number of Raw Materials = 4
- Number of Affiliates = 5
- Number of Grades = 5
- Number of Warehouses = 4
- Number of Customers = 10
- Number of Periods = 8

The data that was used in this example is randomly generated from uniform distribution according to the following limits.

- Transportation cost from supplier *s* to affiliate *a*: U [1, 10] AED.
- Transportation cost from affiliate *a* to warehouse *w*: U [1, 10] AED.
- Transportation cost from warehouse *w* to customer *k*: U [5, 15] AED.
- Purchasing Cost of raw material *r* from supplier *s* to affiliate *a*: U [4, 8] AED.
- Profit of grade *g* in affiliate *a* during time period *t*: U [500, 1000] AED.
- Transitional cost between grade *g* and *h* in affiliate *a* during time period *t*.

  These costs were generated using the following procedure:

  For simplicity, the absolute difference is considered in generating the transitional matrix i.e. a transition from grade one to three is equal to a transition from grade three to one. If the difference between *g* and *h* is 1, the costs follow U [10, 50] AED, if the difference between *g* and *h* is 2, the costs follow U [50, 100] AED, if the difference between *g* and *h* is 3, the costs follow U [100, 150] AED and if the difference between *g* and *h* is 4, the costs follow U [150, 200] AED.

- Inventory carrying cost of grade *g* in warehouse *w* during time period *t*: U [1, 10] AED.
- Availability of raw material *r* from supplier *s*: Constant number of 10^3.
- Production capacity of affiliate *a* during time period *t*: Constant number of 5500.
- Transitional material between grade *g* and *h* in affiliate *a* during time period *t*.

  These amounts were generated in a special procedure. For simplicity, the absolute difference is considered in generating the transitional matrix i.e. a transition from grade one to three is equal to a transition from grade three to one. If the difference between *g* and *h* is 1, the amount follow U [1, 5], if the difference between *g* and *h* is 2, the amount follow U [5, 10], if the difference between *g* and *h* is 3, the amount follow U [10, 15] and if the difference between *g* and *h* is 4, the amount follow U [15, 20].

- Quantity needed of raw material *r* to produce grade *g* in affiliate *a*: U [0, 5].
- Demand of grade *g* for customer *k* during time period *t*: U [100, 1000].
- Percentage of the excess capacity allocated to grade *g* at each period *t*. it was assumed in this example that no excess capacity should be allocated to any grade, therefore the percentage is zero across all grades and periods.

The model was coded using Lingo Software version 15.0 and the developed example was solved. An interface was established between Lingo and Excel to import the data of the parameters and export the data of the decision variables. Lingo code is presented in Appendix A.

Lingo was able to solve the example and an optimal solution was found in a running time of three hours, fifty minutes and twenty Seconds (3:50:20) with a total profit of 170,004,400 AED. The total number of variables is 8807 and the total number of constraints is 8094.

There are six sets of decision variables that the model should solve: the first one is the transportation quantity from suppliers to affiliates, the second is the transportation quantity from the affiliates to the warehouses, third is the transportation quantity from the warehouses to the customers, fourth is the inventory level of each warehouse and the most important two are the production lots in each affiliate along with their sequence. Tables 3 through 6 show a sample of the first four main sets of decisions variables.

Table 3 shows all the amounts of raw materials transported from supplier one (S1) to all affiliates (a1, a2, a3, a4, a5). For example, S1 a4 r3 t1 represents the transported amount of raw material three from supplier one to affiliate four at period one.

Table 3 Sample of amounts transported from supplier 1 to all affiliates

			T1	T2	T3	T4	T5	T6	T7	T8
S1	a1	r1	0	0	0	0	0	0	0	0
		r2	0	0	0	0	0	0	0	0
		r3	0	0	0	0	0	0	0	0
		r4	0	0	0	0	0	0	0	0
	a2	r1	16500	16464	13952	25416	16482	5495	27450	6652
		r2	0	0	0	0	0	0	0	0
		r3	0	0	0	0	0	0	0	0
		r4	0	0	0	0	0	0	0	0
	a3	r1	27500	5496	20893	11000	27455	16497	5495	10998
		r2	0	0	0	0	0	0	0	0
		r3	0	0	0	0	0	0	0	0
		r4	0	0	0	0	0	0	0	0
	a4	r1	12774	8463	5500	16485	5487	2043	16479	5464
		r2	0	0	0	0	0	0	0	0
		r3	14568	17974	16500	21980	3111	0	21972	0
		r4	0	0	0	0	0	0	0	0

Table 3 Sample of amounts transported from supplier 1 to all affiliates (Continued)

			T1	T2	T3	T4	T5	T6	T7	T8
S1	a5	r1	13356	19890	5492	10994	21492	8226	6649	10074
		r2	0	0	0	0	0	0	0	0
		r3	4316	23301	10984	21988	17035	16452	11698	5037
		r4	5497	5493	16476	5497	5373	11008	14967	5037

Table 4 illustrates all the amounts that are produced in affiliate five (A5) and transported to all warehouses (w1, w2, w3, w4). For example, A5 w3 g3 t5 represents the amount of finished goods of grade three produced at affiliate five and transported to warehouse three at period five.

Table 5 displays the amounts of finished goods that are transported from warehouse two (W2) to the first five customers (k1, k2, k3, k4, k5). For example, W2 k2 g2 t5 represents the amount of finished goods of grade 2 transported from warehouse two to customer two at period five.

Table 6 indicates the inventory level of finished goods at the warehouses (W1, W2, W3, W4). For example, W4 g3 t6 represents the inventory level of grade three in warehouse four at the end of period six.

Table 4 Sample of amounts transported from affiliate 5 to all warehouses

			T1	T2	T3	T4	T5	T6	T7	T8
A5	w1	g1	0	0	0	0	0	0	1453	0
		g2	0	0	0	1491	0	0	0	0
		g3	0	390	0	0	0	0	0	0
		g4	435	0	0	0	0	0	0	0
		g5	976	0	0	0	0	0	0	1926
	w2	g1	0	0	5492	0	0	2758	2595	0
		g2	0	0	0	0	0	0	0	0
		g3	0	1382	0	0	1695	0	0	0
		g4	0	0	0	0	0	0	0	0
		g5	2332	1041	0	0	0	0	0	1748
	w3	g1	0	0	0	0	0	0	0	0
		g2	0	0	0	4006	0	2734	556	0
		g3	0	812	0	0	586	0	0	0
		g4	746	0	0	0	1337	0	200	0
		g5	0	0	0	0	0	0	0	0
	w4	g1	0	0	0	0	0	0	689	0
		g2	0	0	0	0	0	0	0	0
		g3	0	1868	0	0	1126	0	0	0
		g4	0	0	0	0	629	0	0	0
		g5	1008	0	0	0	0	0	0	1363

Table 5 Sample of amounts Transported from warehouse 2 to the first 5 customers

			T1	T2	T3	T4	T5	T6	T7	T8
W2	k1	g1	0	0	0	0	0	0	0	0
		g2	751	416	291	0	743	453	544	165
		g3	0	0	0	529	0	0	0	0
		g4	148	491	554	614	232	997	719	0
		g5	634	0	0	0	0	575	0	345
	k2	g1	0	0	0	0	0	0	0	0
		g2	475	596	389	0	344	0	798	845
		g3	0	0	0	329	0	0	0	0
		g4	373	746	300	283	0	0	800	0
		g5	0	0	0	0	0	0	0	0
	k3	g1	0	0	216	0	0	0	611	0
		g2	953	434	142	0	0	0	665	0
		g3	200	771	349	707	824	161	199	730
		g4	330	0	190	870	0	0	0	0
		g5	730	502	463	900	370	793	237	193
	k4	g1	0	352	782	528	162	201	924	259
		g2	241	321	555	0	0	0	583	0
		g3	0	0	0	609	0	270	0	0
		g4	698	346	0	521	172	0	816	0
		g5	0	0	0	0	0	758	0	0
	k5	g1	0	0	527	0	190	318	110	0
		g2	916	0	0	0	0	0	0	0
		g3	0	0	0	824	0	0	0	0
		g4	0	0	0	0	0	0	0	0
		g5	339	0	0	0	0	519	0	0

Table 6 Inventory Levels at the warehouses

		T1	T2	T3	T4	T5	T6	T7	T8
W1	g1	204	0	0	0	2536	0	0	0
	g2	0	1741	0	0	0	0	0	0
	g3	0	0	0	0	1200	1200	914	0
	g4	0	0	0	0	1787	2461	0	0
	g5	0	0	0	151	258	0	0	0
W2	g1	0	0	1723	918	0	0	0	0
	g2	0	0	0	0	0	0	0	0
	g3	0	0	0	0	0	275	0	0
	g4	0	1639	0	0	0	0	0	0
	g5	0	0	0	785	0	0	0	0
W3	g1	0	0	676	0	0	0	0	0
	g2	1616	1616	0	0	0	0	0	0
	g3	0	0	0	0	0	634	9	0
	g4	0	0	0	0	968	0	0	0
	g5	0	0	0	2986	0	0	0	0

Table 6 Inventory Levels at the warehouses (Continued)

		T1	T2	T3	T4	T5	T6	T7	T8
W4	g1	0	0	2398	0	0	0	0	0
	g2	0	0	0	0	0	0	0	0
	g3	20	0	0	0	3554	3399	628	0
	g4	0	0	0	0	0	0	0	0
	g5	0	0	0	0	0	0	0	0

More details will be given on the last two sets of decision variables because of their importance. These two decisions are the lot size to be produced in each affiliate and the sequence of the production lots in each affiliate.

- Affiliate one

Table 7 shows the amounts of each grade that should be produced in affiliate one, while Table 8 represents the sequence of the corresponding production plan.

Table 7 Production plan of affiliate 1

		T1	T2	T3	T4	T5	T6	T7	T8
A1	g1	0	5278	4618	0	5167	0	0	0
	g2	5299	0	0	0	321	0	263	138
	g3	200	216	0	5498	0	0	0	0
	g4	0	0	872	0	0	0	0	5353
	g5	0	0	0	0	0	5489	237	0

Table 8 Production plan sequence of affiliate 1

	T1	T2	T3	T4	T5	T6	T7	T8
A1	2 - 3	3 -1	1 - 4	4 - 3	3 - 1 - 2	2 - 5	5 - 2	2 - 4

- Affiliate two

Table 9 illustrates the amounts of each grade that should be produced in affiliate two, while Table 10 represents the sequence of the corresponding production plan.

Table 9 Production plan of affiliate 2

		T1	T2	T3	T4	T5	T6	T7	T8
A2	g1	5500	0	0	0	0	0	0	583
	g2	0	0	2116	793	0	0	0	0
	g3	0	0	3372	137	0	5495	0	4903
	g4	0	5488	0	738	5494	0	0	0
	g5	0	0	0	3820	0	0	5490	0

Table 10 Production plan sequence of affiliate 2

	T1	T2	T3	T4	T5	T6	T7	T8
A2	1	1 - 4	4 - 2 - 3	3 - 5 - 4 -2	2 - 4	4 - 3	3 - 5	5 - 3 - 1

From Table 9, we can conclude that in affiliate two (A2), at period four (T4), a production must be done for grade two (g2), grade three (g3), grade four (g4) and grade five (g5) with a sequence of grade three (g3), grade five (g5), grade four (g4) and grade two (g2) from Table 10.

- Affiliate three

Table 11 indicates the amounts of each grade that should be produced in affiliate three, while Table 12 shows the sequence of the corresponding production plan.

Table 11 Production plan of affiliate 3

		T1	T2	T3	T4	T5	T6	T7	T8
A3	g1	0	0	0	0	0	0	0	5499
	g2	0	5496	0	0	0	0	5495	0
	g3	5500	0	2329	0	5491	0	0	0
	g4	0	0	2922	0	0	5499	0	0
	g5	0	0	241	5500	0	0	0	0

Table 12 Production plan sequence of affiliate 3

	T1	T2	T3	T4	T5	T6	T7	T8
A3	3	3 - 2	2 - 3 - 4 - 5	5	5 - 3	3 - 4	4 - 2	2 - 1

- Affiliate four

Table 13 represents the amounts of each grade that should be produced in affiliate four, while Table 14 illustrates the sequence of the corresponding production plan.

Table 13 Production plan of affiliate 4

		T1	T2	T3	T4	T5	T6	T7	T8
A4	g1	0	0	0	0	0	0	0	0
	g2	1848	0	0	0	4450	2043	0	5464
	g3	0	0	0	0	0	0	0	0
	g4	3642	1483	0	5495	0	0	5493	0
	g5	0	4014	5500	0	1037	0	0	0

Table 14 Production plan sequence of affiliate 4

	T1	T2	T3	T4	T5	T6	T7	T8
A4	2 - 4	4 - 5	5	5 - 4	4 - 5 - 2	2	2 - 4	4 - 2

- Affiliate five

Table 15 represents the amounts of each grade that should be produced in affiliate five, while Table 16 displays the sequence of the corresponding production plan.

Table 15 Production plan of affiliate 5

		T1	T2	T3	T4	T5	T6	T7	T8
A5	g1	0	0	5492	0	0	2758	4737	0
	g2	0	0	0	5497	0	2734	556	0
	g3	0	4452	0	0	3407	0	0	0
	g4	1181	0	0	0	1966	0	200	0
	g5	4316	1041	0	0	0	0	0	5037

Table 16 Production plan sequence of affiliate 5

	T1	T2	T3	T4	T5	T6	T7	T8
A5	4 - 5	5 - 3	3 - 1	1 - 2	2 - 4 -3	3 - 2 - 1	1 - 2 - 4	4 - 5

The outcome of the model regarding the sequence in each affiliate is given in binary variables form. Table 17 presents an example of sequence output for affiliate one.

Table 17 Sample of optimal sequence of affiliate 1

		T1	T2	T3	T4	T5	T6	T7	T8
A1	1 to 1	0	0	0	0	0	0	0	0
	1 to 2	0	0	0	0	1	0	0	0
	1 to 3	0	0	0	0	0	0	0	0
	1 to 4	0	0	1	0	0	0	0	0
	1 to 5	0	0	0	0	0	0	0	0
	2 to 1	0	0	0	0	0	0	0	0
	2 to 2	0	0	0	0	0	0	0	0
	2 to 3	1	0	0	0	0	0	0	0
	2 to 4	0	0	0	0	0	0	0	1
	2 to 5	0	0	0	0	0	1	0	0
	3 to 1	0	1	0	0	1	0	0	0
	3 to 2	0	0	0	0	0	0	0	0
	3 to 3	0	0	0	0	0	0	0	0
	3 to 4	0	0	0	0	0	0	0	0
	3 to 5	0	0	0	0	0	0	0	0

Table 17 Sample of optimal sequence of affiliate 1 (Continued)

		T1	T2	T3	T4	T5	T6	T7	T8
	4 to 1	0	0	0	0	0	0	0	0
	4 to 2	0	0	0	0	0	0	0	0
	4 to 3	0	0	0	1	0	0	0	0
	4 to 4	0	0	0	0	0	0	0	0
	4 to 5	0	0	0	0	0	0	0	0

The most important outputs from lingo are the production plan and sequence for each affiliate. The model will maximize the total profit by deciding on grades and lot size in each affiliate in each time period while considering the production sequence of these lots. As mentioned earlier, the sequence of these lots will affect the total profit since each sequence has its own transitional cost. At the same time, each sequence has its own transitional amount that affects the available capacity at each period. It can be seen that the model will take advantage of carry-over setup which means that the last grade produced in each period is the same grade produced at the beginning of the next period.

The size of this problem is considered a medium size problem, therefore Lingo 15.0 was able to solve it in medium range of time. When the size of the problem increases, Lingo 15.0 will not be able to solve these problems in a realistic limited time. Therefore a Heuristic is needed to be developed in order to solve the model in a reasonable time with good quality solution. The following section illustrates the heuristic proposed to solve the problem in this research.

3.6 Heuristic Solution Approach:

As mentioned earlier, the capacitated lot-sizing and scheduling problem is NP-Hard, thus commercial optimization software will not solve the CLSP problem in a reasonable time. In this section, a three-stage heuristic is proposed to solve the CLSP with sequence-dependent cost and quantity. Figure 9 summarizes the heuristic stages: the first stage is the transformation from CLSP to CLP with sequence-independent setup costs, the second stage is the sequence reduction and optimization that will reduce the total setup cost by reducing the number of setups and the third stage is the inventory reduction that will reduce the total holding cost by reducing the cumulative inventory. The heuristic were coded using Matlab 15.0. A sample of the code can be found in Appendix C.

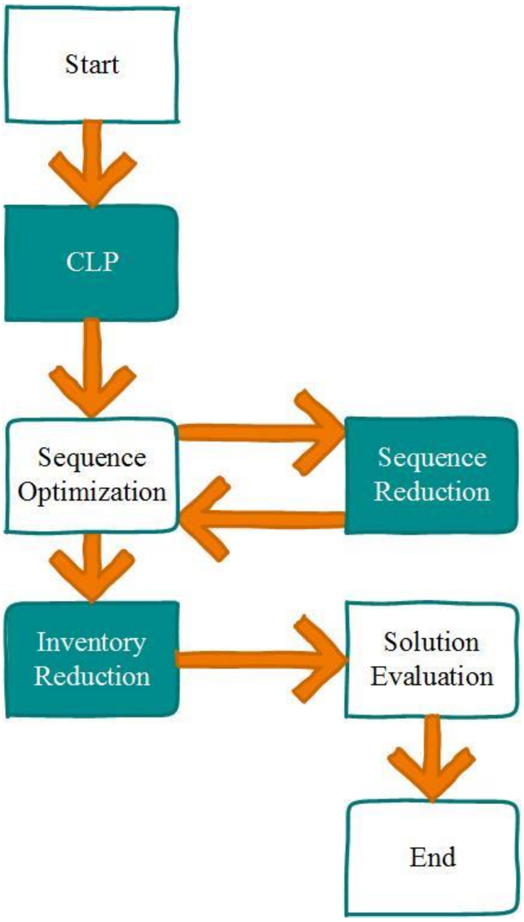

Figure 9 Three-Stage Heuristics

3.6.1 Stage one: capacitated lot-sizing problem with sequence independent. According to Florian et al. [48] CLSP without the sequence-dependent setup costs is proven to be NP-Hard, the sequencing problem with sequence-dependent setup by itself is also proven to be NP-Hard by Coleman [49]. Combining these two problems resulted in having a CLSP with sequence-dependent setup costs that are NP-Hard. This stage is concerned with reducing the complexity of the CLSP with sequence-dependent setup by transforming the problem to Capacitated Lot-sizing with sequence-independent setups (CLP). In this case, the first stage will decide the size of the production lots only, while the other stages will consider the sequence.

CLP formulation is similar to the developed model of CLSP at the beginning of this section, it has the same indices, parameters and decision variables except the following:

- Different Parameters

$\boldsymbol{C_{agt}^{Setup}}$ Setup cost of grade *g* when it is produced in affiliate *a* during time period *t*.

$\boldsymbol{O_{agt}}$ Transitional material formed per ton when grade *g* is produced in affiliate *a* during time period *t*.

Note that this model does not have the sequence-dependent and the excess capacity parameters ($\boldsymbol{C_{aght}^{nprime}}$, $\boldsymbol{O_{aght}}$, $\boldsymbol{EX_t}$, $\boldsymbol{\rho_{gt}}$).

- Different Decision Variable

$\boldsymbol{Y_{agt}}$ A binary variable that equals 1, if grade *g* is produced in affiliate *a* during time period *t*, 0 otherwise.

Note that this model does not have the sequence-dependent and the sub-tour elimination decision variables ($\boldsymbol{Z_{aght}}$, $\boldsymbol{\alpha_{agt}}$, $\boldsymbol{V_{agt}}$).

- Model Formulation
  - ***Objective Function***

Using the same objective function (1) with a minor difference where the fourth part is modified from sequence-dependent setup cost to sequence-independent setup cost when different grades are being produced. The following part is used.

$$\left[\sum_{a=1}^{A}\sum_{g=1}^{G}\sum_{h=1,h\neq g}^{H}\sum_{t=1}^{T} C_{agt}^{Setup} * O_{agt} * Y_{agt}\right] \tag{16}$$

- ***Supplier Constraints***

Constraints (2) and (3) are used for the supplier constraints.

- ***Affiliate Constraints***

$$\sum_{g=1}^{G} X_{agt} + \sum_{g=1}^{G} O_{agt} * Y_{agt} \leq \lambda'_{at} \qquad \forall\, a\,, t \tag{17}$$

The total amount produced at each period *t* of prime grades and transitional material in each affiliate *a* cannot exceed its capacity.

$$X_{agt} \leq M * Y_{agt} \qquad \forall\, a, g, t \tag{18}$$

Constraint (18) guarantees that grade g in each affiliate a can be produced at period t only if the machine is setup for the grade.

- ***Warehouse Constraints***

Constraints (12) and (13) are used for the warehouse constraints.

- ***Customer Constraints***

$$\sum_{w=1}^{W} F''_{wkgt} = D_{gkt} \qquad \forall\, g, k, t \tag{19}$$

Constraint (19) indicates that the total amount of grade g transported from all warehouses w should be equal to the demand of each customer k at each period t.

Sequence-independent setup costs and quantities need to be estimated to be able to run the model. According to Kwak and Jeong [13] the sequence-dependent setup costs can be transformed to sequence-independent setup costs by applying the following formula.

$$C_{agt}^{Setup} = \begin{cases} \dfrac{C_{a(2)(g)t}^{Setup} + C_{a(3)(g)t}^{Setup}}{2}, & g = 1 \\ C_{a(g+1)(g)t}^{Setup}, & g = 2, \dots, G-1 \\ \dfrac{C_{a(1)(g)t}^{Setup} + C_{a(2)(g)t}^{Setup}}{2}, & g = G \end{cases} \tag{20}$$

As for the sequence-dependent setup quantity, the sequence independent quantity will be considered as the max of the sequence-dependent quantity. In this case, if the maximum quantity results from a transition between grade 1 and grade 5, the sequence independent quantity for all grades will be considered equal to the transitional material between grade 1 and grade 5. This assumption is made to ensure that the initial plan from the first stage is feasible in terms of the capacity. This provides the flexibility to the next stage in sequence optimization to have at least more than one feasible sequence.

3.6.2 Stage two: sequence reduction and optimization.

3.6.2.1 Sequence optimization. In this stage, an initial plan will be received from the CLP model. The first phase will be to optimize the sequence of the initial plan because the total transitional material amount will be used in the sequence reduction step.

The sequence optimization phase starts by considering all sequence possibilities in each affiliate. At each period, certain products are going to be produced, the algorithm will compute the permutations of the sequence of these products. A permutation is used because it provides all the different sequence possibilities where the sequence order matters. For example in period t, product A and product C are produced, the sequence permutations will be Product A then Product C or Product C then Product A. This step is made for each period then these alternatives are combined in a certain arrangement to provide all sequence possibilities that can occur in each affiliate starting from period t to period T. Due to the carry-over setup option, periods in each affiliate are connected where the last product produced in period t and the first product produced in period t +1 can affect the total transitional cost.

Figure 10 illustrate the steps of sequence optimization. The following is the detailed steps of this phase:

Step one.

a) Set $a = 1$. If $a > A$, End.
b) Set $t = 1$. If $t > T$, move to step one (a).
c) Set $g = 1$. If $g > G$, move to step one (b).
d) Check if grade g is greater than zero, if yes move to step one (e), otherwise, move to step one (c).
e) Calculate the permutation of all grades at period t. if grade g is greater than G, move to step one (b), otherwise, move to step one (c).
f) Check if period t is greater than T, if yes move to step two, otherwise, move to step one (b).

Step two.

g) Arrange all permutations of all periods to get all possible routes.
h) Check feasibility by calculating the total transitional material and comparing it with the available capacity.
i) Calculate the total transitional cost of each route.
j) Choose the route with the least cost. Go to step one (a).

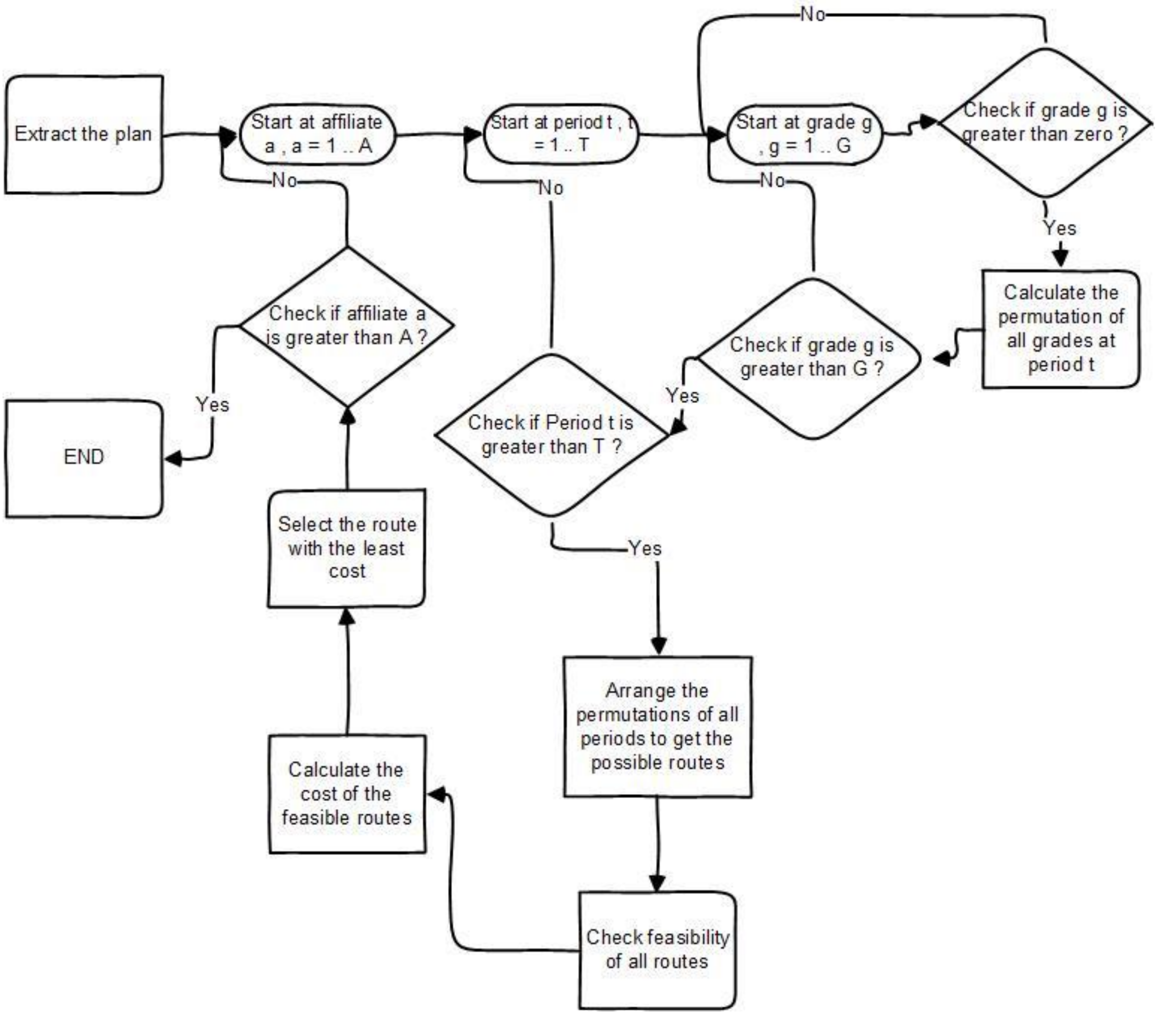

Figure 10 Sequence Optimization

Figure 11 illustrates an example of the possibilities of the sequences in a certain affiliate. In period *t*, Product A and C are produced, in period *t*+1, Product B is produced and in period *t*+2, Product A and B are produced. It can be seen that four different routes can be obtained from the algorithm. This is expected because the number of routes follows a mathematical formula where it is equal to the multiplication of the number of permutations of each period. In the previous example, in period *t*, 2 permutations can be obtained, 1 in period *t*+1 and 2 in period *t*+2. The multiplication of these permutations is 2*1*2 = 4 different routes.

After computing the different routes, the algorithm will evaluate each route in terms of cost and feasibility. The cost will be calculated based on the transitional matrix cost and the feasibility will be tested based on the transitional quantity that each route will produce. If the transitional quantity beside the production amount is less than the available capacity, the route will be considered feasible, otherwise, it will be unfeasible. Finally, the algorithm will choose the lowest cost feasible route.

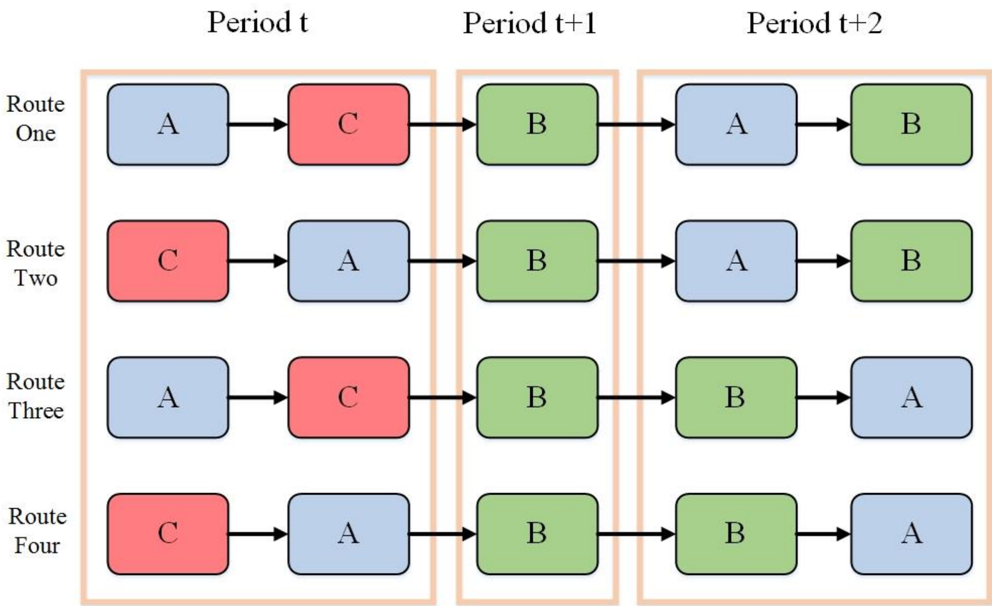

Figure 11 Sequence Optimization Routes

3.6.2.2 Sequence Reduction. In this phase, the algorithm will try to reduce the number of setups. This phase starts from the last period and ends at the first period which can be justified since the algorithm will reduce the number of setups by pushing back the amount of a certain grade to be produced to the previous period. In this case, the number of setups will be reduced, no shortages will occur since the required amount of that grade will be satisfied from the inventory and the inventory cost will be increased. This increment will be reduced in the third stage.

The movement of the grades can be done from period t to period t-1 in the same affiliate or to a different affiliate based on certain rules that will be explained in the detailed steps. If a movement was done, this means an available capacity was created at that period, the model will start over from the beginning to use that available capacity in other movements.

There are two options in this algorithm, the first option is to fix the affiliate number and vary the time and when the time is over (t=1) it moves to the next affiliate. On the other hand, option two is to fix the time and vary the affiliate number and when all affiliates are visited by the algorithm, it moves to the next period (t = t-1). This is done to increase the number of the outputs, where these outputs might be different due to the fact that the movement strategy is different.

Figure 13 and Figure 14 illustrate the steps of each option in this phase which can be described as follow:

Stage two (reduction of the number of setups).

Step one.

Option one:

a) Set $L = a$. if $L > A$, Move to stage three.
b) Set $m = T$.
c) Set $n = m$-1. If $m = t$, increment L.

Option two:

a) Set $m = T$. if $m = t$, Move to stage three.
b) Set $n = m$-1.
c) Set $L = a$. if $L > A$, decrement m.

Step two.

d) Select a grade g from period m using priority rule. If no grade were produced, decrement m in option one and increment L in option two. Go to step two.

Step three.

e) Check if the amount of the selected grade g is less than the available capacity at period n. if not go to step four.
f) Check if in period n the same grade g was produced. If not go to step four.
g) Move grade g from period m to period n.
h) Update the sequence.
i) Set m to m+1 in option one and option two. Go to step two.

Step four.

j) Check if the amount of the selected grade g is less than the available capacity at period n in at least one affiliate. If not Decrement m in option one and increment L in option two. Go to step two.
k) Check if in period n the same grade g was produced in at least one affiliate. If not Decrement m in option one and increment L in option two. Go to step two.

l) Choose the affiliate with the highest profit.

m) Move grade *g* to the chosen affiliate from period *m* to period *n*.

n) Update the sequence. Set *m* to *m*+1 and set *L* to 1 in option one and two. Go to step two.

When a move is done from one period to the previous one in the same affiliate, then index *m* is restored to *m*+1. Also, whenever a move is done from one period to a previous one in another affiliate, then *m* is restored to *m*+1 and index *L* is restored to 1. This procedure is done to include the following special case illustrated in Figure 12.

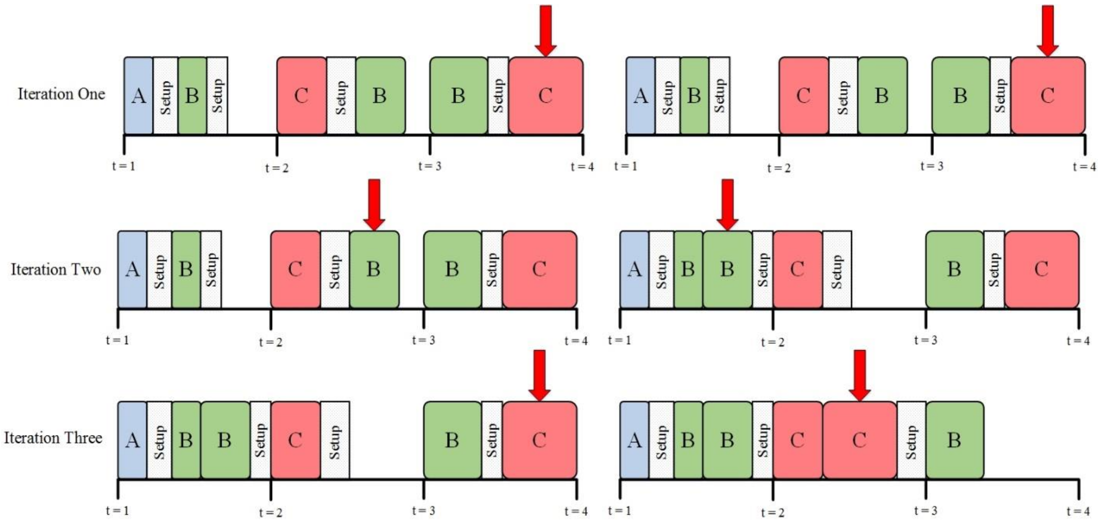

Figure 12 Stage two Special Case

In Iteration one, grade C at period 3 is chosen to be moved, the amount of grade C is greater than the available capacity at period 2, therefore nothing will be moved in this period. In iteration two, grade B at period 2 is chosen to be moved, it can be moved because the available capacity is equal to grade B amount and the same grade is produced at period 1, therefore grade B will be moved from period 2 to period 1. The movement of grade B at period 2 increased the available capacity at the same period. In order to take advantage of this available capacity, the algorithm will go back to period 3 and go through iteration three where grade C will be moved from period 3 to period 2 since the available capacity is greater than grade C and the same grade is being produced at period 2. The same concept applies when the movement is done to different affiliate.

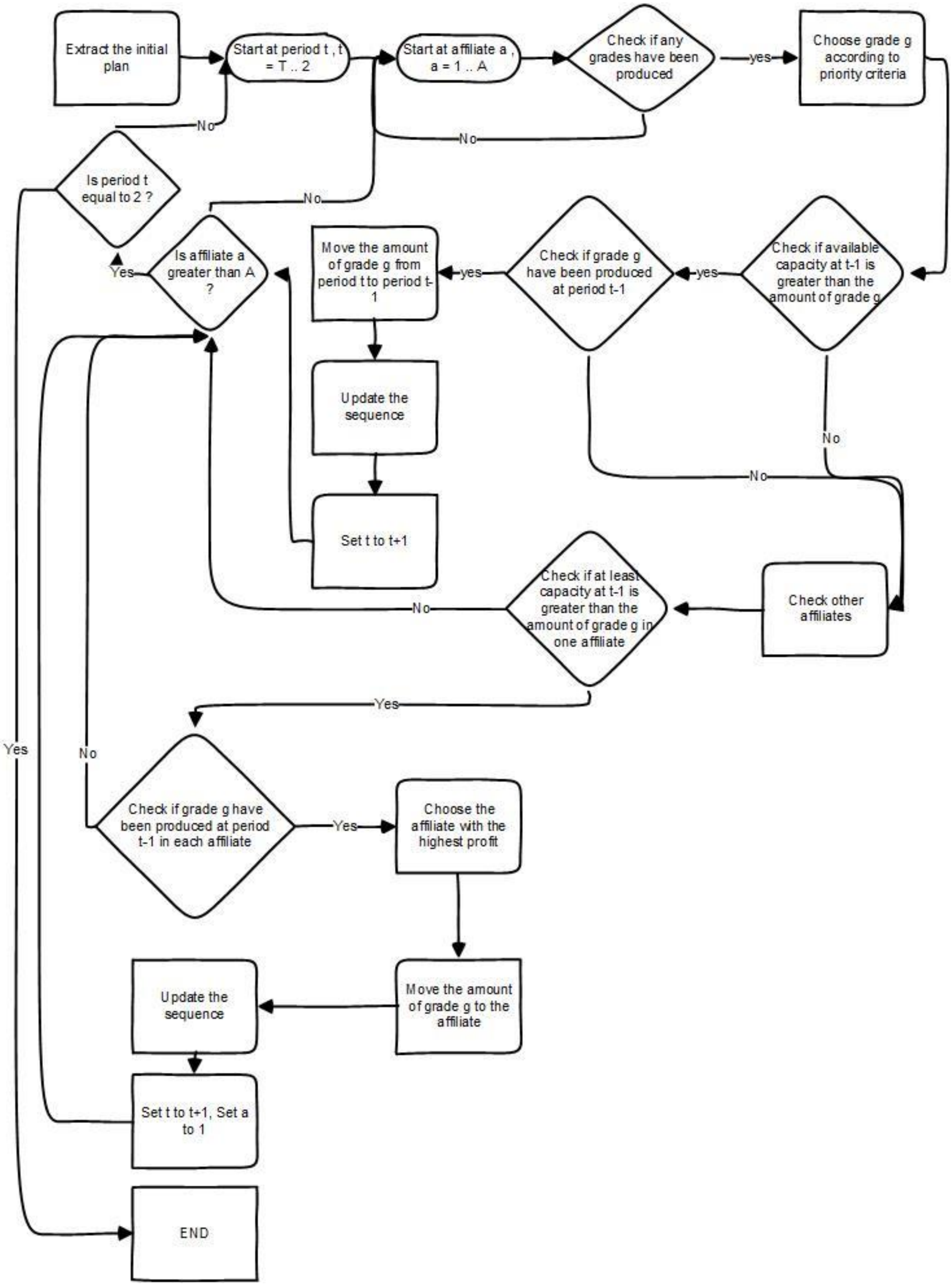

Figure 13 Stage two option one

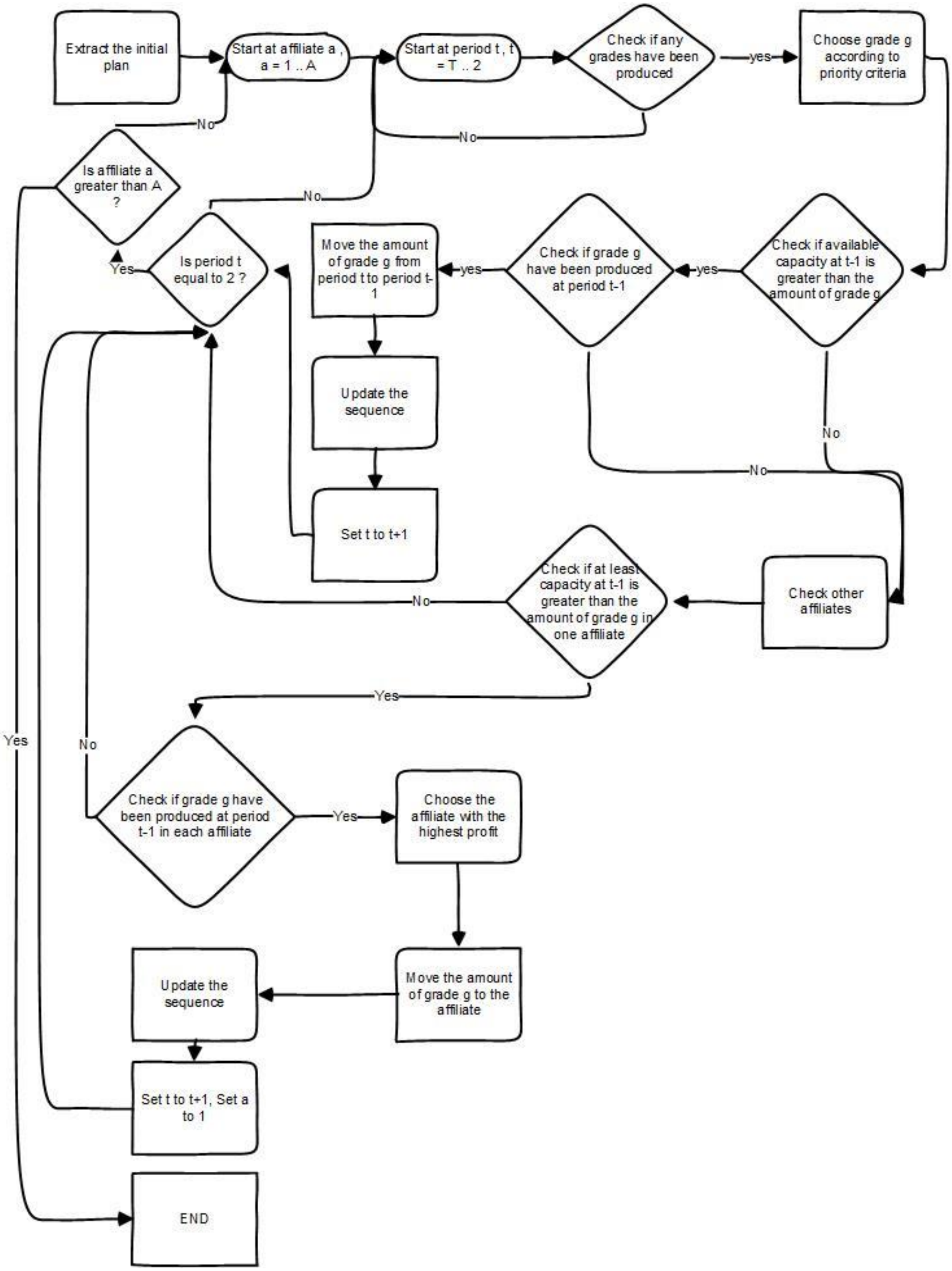

Figure 14 Stage two option two

Three scenarios can happen in stage two which are:

- Scenario one.

In any scenario, a grade is chosen to be moved from period t to period t-1 in affiliate a, the algorithm will check if the same grade is being produced in period t-1 or not. Then it will check if the amount of the selected grade is less than the available capacity or not. In this scenario, the affiliate is producing the same grade in the previous period and there is enough capacity to move the amount from period t to period t-1; Figure 15 illustrates this case. Grade C at period 3 is chosen to be moved to the previous period. The same grade is being produced at period 2 and there is enough capacity to shift grade C from period 3 to period 2. In this case, the algorithm will shift the grade to the previous period without considering other affiliates because the two main conditions of shifting are satisfied.

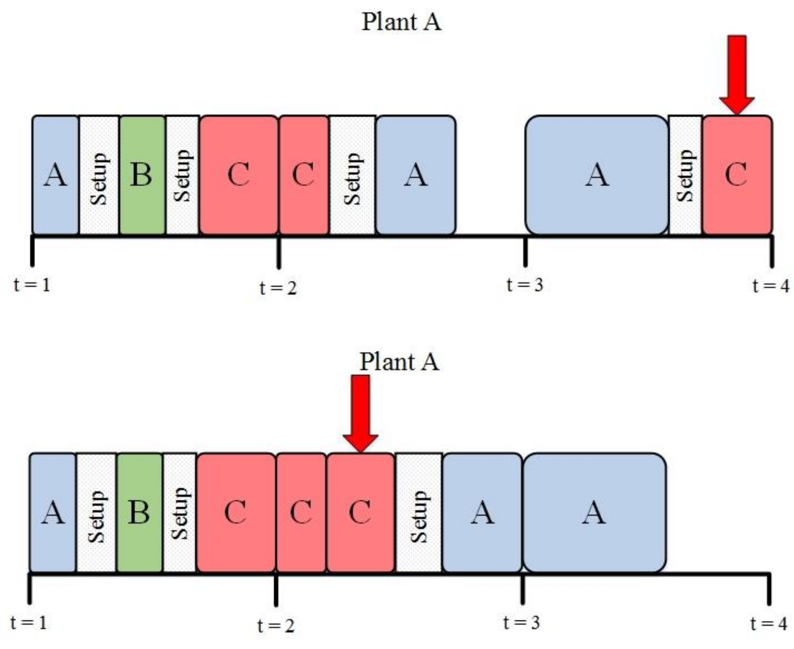

Figure 15 Scenario one in stage two

- Scenario two.

In this scenario, the affiliate is producing the same grade in the previous period, however, there is not enough capacity to move the grade to the previous period. The algorithm will start looking for other affiliates that satisfy the main two conditions and will move the grade to the affiliate with the highest profit. Figure 16 shows an example on this case. Grade C in plant A is chosen to be moved from period 3 to period 2. At period 2, plant A is producing the same grade however there is no enough capacity to accommodate grade C from period 3, therefore other plants will be checked such as plant B and plant C. it can be seen that plant B is the only one that is producing grade C and has enough capacity at period 2, if more than one plant satisfy these two conditions, the one with the highest profit will be chosen to receive the movement. The algorithm will move grade C at period 3 from plant A to plant B at period 2.

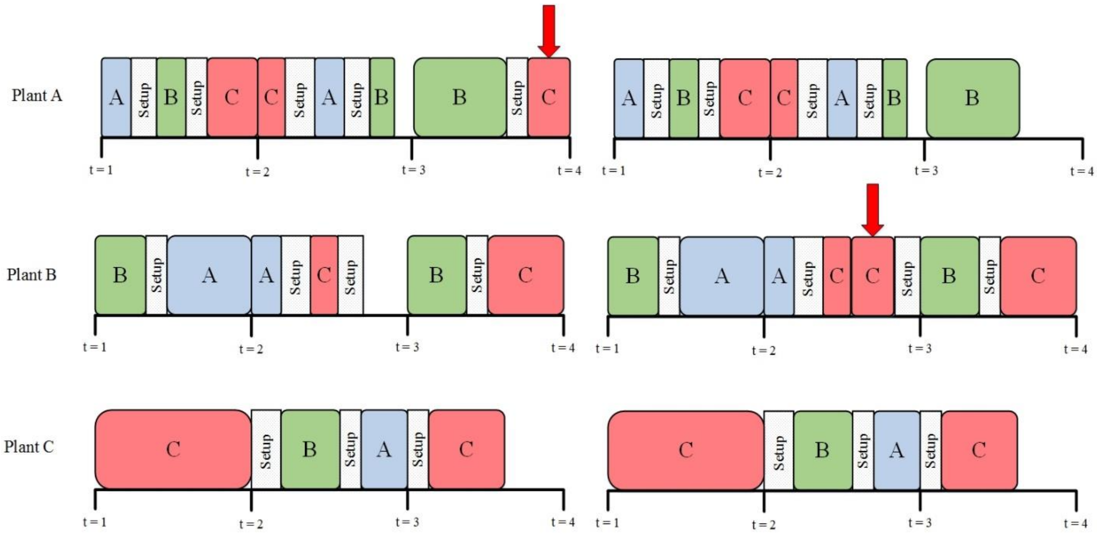

Figure 16 Scenario two in stage two

- Scenario three.

In the last scenario, the affiliate has enough capacity to accommodate the movement of the grade from the preceding period, however, it isn't producing the same grade in that period. The algorithm will start looking for other affiliates that satisfy the two main conditions and will move the grade to the affiliate with the highest profit. Figure 17 shows an example of this case. In Figure 17, grade C in plant A is chosen to be moved from period 3 to period 2. At period 2, plant A has enough capacity to accommodate the movement of grade C from period 3, however, it doesn't produce the same grade. Therefore, the algorithm will search for other plants, B and C in this example. Since they satisfy the two conditions in this scenario. As a result, the algorithm will choose the plant that generates the highest profit from producing grade C at period 2. Plant C is chosen to move grade C from plant A from period 3 to period 2.

There is an iterative relation between the sequence reduction and the sequence optimization phases. Whenever a movement is done, the system will call the sequence optimization function to optimize the sequence. It is important to know the sequence of the plan after the shift to calculate the total transitional amount because it is sequence dependent. This total transitional amount affects the available capacity in each period which consequently affects the movements of the grades between periods.

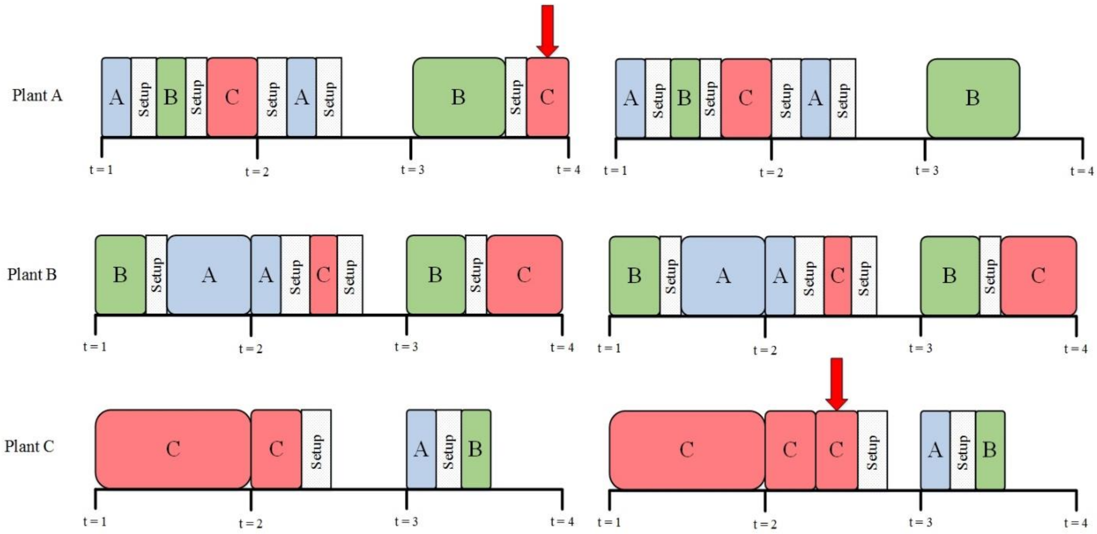

Figure 17 Scenario three in stage two

3.6.3 Stage three: inventory reduction. The goal of this stage is to reduce the total cumulative inventory in each period. After the reduction in sequence from stage two, the inventory of the grades was increased in some of the periods due to the pushing back of the grades from period T to any of the previous periods. This is expected because the second stage algorithm increases the production amounts in previous periods in order to reduce the number of setups. In other words, later periods demand can be satisfied from the inventory because the required amounts were produced earlier and kept in the inventory. Therefore this stage pushes forward the cumulative inventory in order to reduce the total amount of holding costs.

The main condition in this stage is not to increase the number of setups. The movement of the inventory will be done only if the same grade is being produced in the later period while satisfying the available capacity limitation. Since stage two has two options, then two outputs will be extracted in this stage. The same steps will be repeated for each output. The algorithm will start by calculating the total production amount in each period for each demand then it will calculate the total demand in each period for each grade. After that, the total cumulative inventory will be calculated based on the following formula.

$$CI_{gt} = TP_{gt} - TD_{gt} + CI_{gt-1} \tag{21}$$

$CI_{gt} =$ *Cumulative inventory of grade g at period t*

$TP_{gt} =$ *Total production of grade g at period t*

$TD_{gt} =$ *Total demand of grade g at period t*

Since the warehouses are outside the affiliates and not considered in this heuristic, it is assumed that the holding cost is similar to each period in each warehouse, therefore this stage considers the total cumulative inventory regardless of the location. After the cumulative inventory is calculated, the algorithm will start pushing these amounts forward to a future time period to reduce it as much as possible.

There are two options in this algorithm, the first option is to fix the time and vary the grade; when the grade is over (g=G), it moves to the next period. On the other hand, option two is to fix the grade and vary the time, when all periods are visited by the algorithm, it moves to the next grade (g = g+1). This is done to increase the number of the outputs, where these outputs might be different due to the fact that the movement strategy will be different. Figure 18 and Figure 19 illustrate the steps for option 1 and option 2 respectively which can be described as follow:

Step one.

a) Set t = 1. If t > T, move to step two.
b) Set g = 1. If g > G, move to step one (a).
c) Set a = 1. If a > A, move to step one (b).
d) Calculate the Total production by adding the amount produced of affiliate a. increment a and move to step one (d).

Step two.

e) Set t = 1. If t > T, move to step three.
f) Set g = 1. If g > G, move to step one (e).
g) Set k = 1. If k > K, move to step one (f).
h) Calculate the Total demand by adding the required amount of customer k. increment k and move to step two (g).

Step three.

i) Subtract the output of step one from the output from step two to get the total cumulative inventory.

Step four.

Option one:

j) Set $t = 1$. If $t > T$, move to end.
k) Set $g = 1$. If $g > G$, move to step four (j).

Option two:

j) Set $g = 1$. If $g > G$, move to end.
k) Set $t = 1$. If $t > T$, move to step four (j).
l) Choose the affiliate that produces grade g with the least profit.
m) Choose the affiliates that have an available capacity that is greater than zero at period t+1 and produce grade g.
n) Select the affiliate that has the highest profit at period t+1.

Step five.

o) Calculate the minimum amount of the cumulative inventory of grade g at period t, the amount produced at the selected affiliate at period t, the available capacity of the selected affiliate at period t+1.
p) Check if the minimum amount is greater than zero. If yes move to step five (r), otherwise move to step five (q).
q) Check if all affiliates were selected. If yes move to step four (k) in both options, otherwise step four (I).
r) Move the minimum amount from/to the chosen affiliates from period t to period t+1.
s) Check if the cumulative amount is greater than zero. If yes move to step four (I), otherwise move to step four (k) in both options.

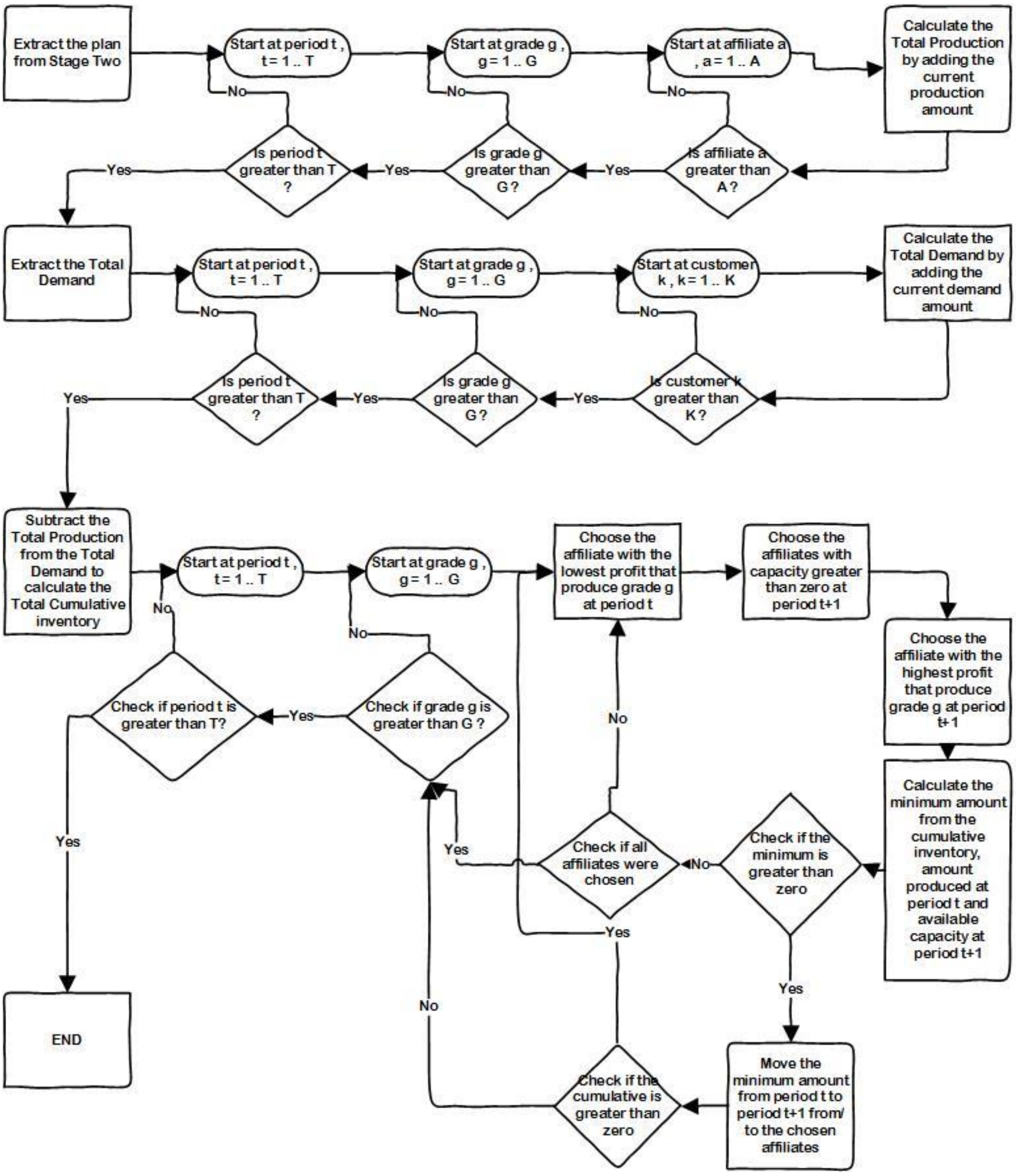

Figure 18 Stage three option one

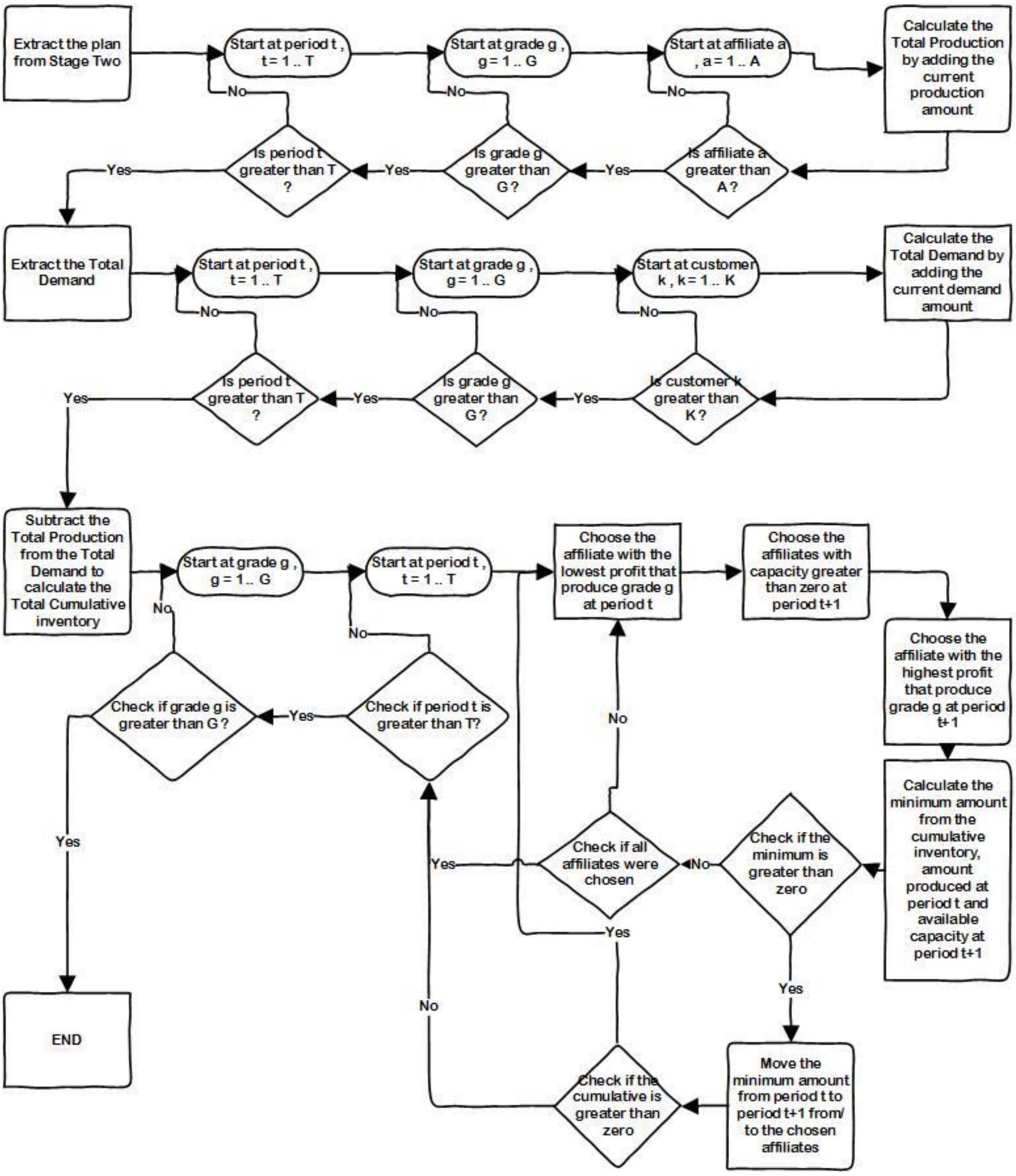

Figure 19 Stage three option two

Stage two has two different outputs from option one and option two. Stage three also has two different outputs from option one and option two. Since stage three depends on the previous stage output, this means four different outputs will be generated from the heuristic. Figure 20 is an illustrative example of the third stage algorithm.

Plant A and Plant B produce two grades A and B. In the first period, the total production from grade A and B is greater than the total demand, therefore a cumulative inventory exists. The algorithm start by choosing grade A to be shifted from Plant A because it has a lower profit than Plant B at period 1. The shift is done to Plant B at period 2 because it has a higher corresponding profit than Plant A at period 2. Then the next grade B is chosen to be shifted from Plant B because it has a lower profit than Plant A at period 1. The shift is done to Plant A at period 2 because it has a higher corresponding profit than Plant B at period 2.

Finally, after the heuristic is done, four different plans will be generated and the final step will be to evaluate these plans in terms of profit and take the one with the maximum profit.

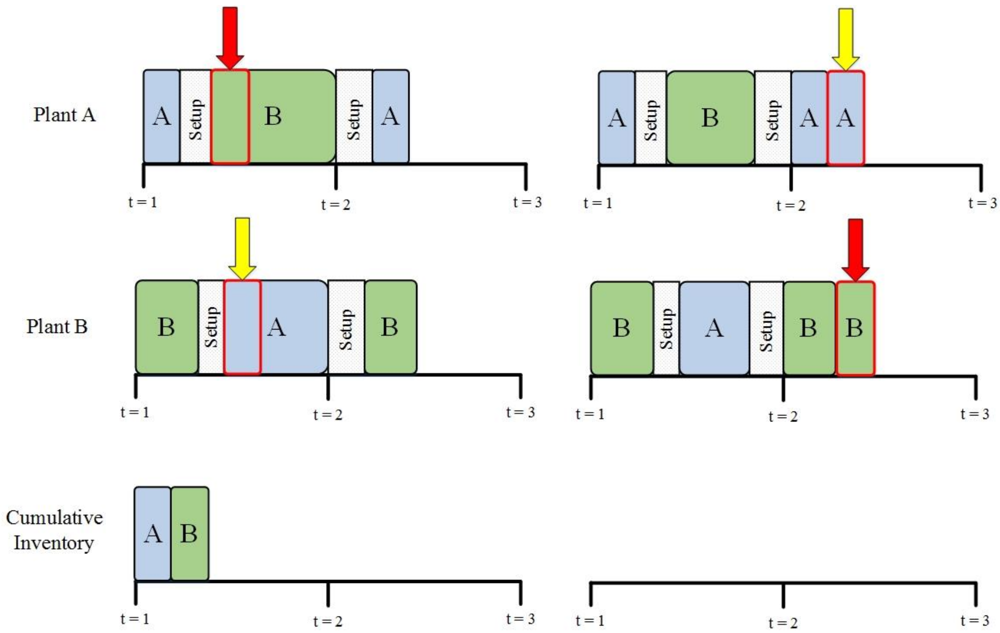

Figure 20 Stage three example

3.7 Illustrative Example

In this section, the same example that was solved previously in the exact method (Section 3.5) will be solved using the proposed heuristic in order to compare the results in terms of computational time and results quality.

3.7.1 Stage one CLP. The first step in the heuristic is to transform the exact formulation from CLSP to CLP in order to reduce the complexity and reduce the computational time. The sequence independent costs and quantities have to be estimated in order to run the model. Table 18 and Table 20 present the estimated costs and quantities.

Table 18 Sequence Independent Estimated Costs

		T1	T2	T3	T4	T5	T6	T7	T8
A1	g1	45	50	55	35	45	35	35	50
	g2	40	10	40	50	40	50	50	10
	g3	40	10	40	20	30	20	20	10
	g4	50	30	10	30	10	30	30	30
	g5	175	140	165	150	130	150	150	140
A2	g1	65	60	40	45	65	45	45	60
	g2	40	20	50	20	50	20	20	20
	g3	20	50	50	30	50	30	30	50
	g4	50	10	20	20	10	20	20	10
	g5	125	140	155	140	150	140	140	140
A3	g1	45	45	45	50	60	45	60	55
	g2	20	20	40	10	20	20	20	20
	g3	30	30	40	10	50	30	50	30
	g4	20	20	50	30	10	20	10	50
	g5	140	140	175	140	140	140	140	140
A4	g1	55	55	40	55	60	55	55	55
	g2	40	10	20	20	40	20	20	10
	g3	40	20	50	30	30	30	30	20
	g4	10	40	30	50	10	50	50	40
	g5	140	160	170	140	150	140	140	160
A5	g1	45	65	45	40	65	40	40	65
	g2	40	40	50	40	50	40	40	40
	g3	50	10	50	10	10	10	10	10
	g4	40	50	30	40	50	40	40	50
	g5	175	135	150	145	155	145	145	135

These costs are estimated based on formula (24). The following example illustrates the procedure of estimating the costs. Table 19 is a sample of the transitional costs that were used in the numerical example.

Table 19 Sample of the Transitional Cost

		T1			T1
A3	1 to 1	1000000	A3	3 to 4	20
	1 to 2	40		3 to 5	50
	1 to 3	50		4 to 1	130
	1 to 4	140		4 to 2	50
	1 to 5	170		4 to 3	30
	2 to 1	10		4 to 4	1000000
	2 to 2	1000000		4 to 5	50
	2 to 3	10		5 to 1	200
	2 to 4	80		5 to 2	150
	2 to 5	110		5 to 3	60
	3 to 1	80		5 to 4	20
	3 to 2	20		5 to 5	1000000

- According to the estimating formula, A3 g1 T1 is equal to (A3 2 to 1 T1 + A3 3 to 1 T1)/2 = (10+80)/2 = 45.
- According to the estimating formula, A3 g3 T1 is equal to A3 4 to 3 T1 = 30.
- According to the estimating formula, A3 g5 T1 is equal to (A3 1 to 5 T1 + A3 2 to 5 T1)/2 = (170+110)/2 = 140.

Table 20 is the transitional quantity table, the values inside are estimated based on the maximum transitional quantity.

Table 20 Sequence Independent Estimated quantities

		T1	T2	T3	T4	T5	T6	T7	T8
A1	g1	20	20	20	20	20	20	20	20
	g2	20	20	20	20	20	20	20	20
	g3	20	20	20	20	20	20	20	20
	g4	20	20	20	20	20	20	20	20
	g5	20	20	20	20	20	20	20	20
A2	g1	20	20	20	20	20	20	20	20
	g2	20	20	20	20	20	20	20	20
	g3	20	20	20	20	20	20	20	20
	g4	20	20	20	20	20	20	20	20
	g5	20	20	20	20	20	20	20	20
A3	g1	20	20	20	20	20	20	20	20
	g2	20	20	20	20	20	20	20	20
	g3	20	20	20	20	20	20	20	20
	g4	20	20	20	20	20	20	20	20
	g5	20	20	20	20	20	20	20	20

Table 20 Sequence Independent Estimated quantities (Continued)

		T1	T2	T3	T4	T5	T6	T7	T8
A4	g1	20	20	20	20	20	20	20	20
	g2	20	20	20	20	20	20	20	20
	g3	20	20	20	20	20	20	20	20
	g4	20	20	20	20	20	20	20	20
	g5	20	20	20	20	20	20	20	20
A5	g1	20	20	20	20	20	20	20	20
	g2	20	20	20	20	20	20	20	20
	g3	20	20	20	20	20	20	20	20
	g4	20	20	20	20	20	20	20	20
	g5	20	20	20	20	20	20	20	20

Based on the transitional quantity amounts that were used in the numerical example, the maximum transitional amount is 20, therefore, the amounts will be 20 for all grades and affiliates.

The model was coded and the example was solved using the optimization software Lingo 15.0. The code can be found in Appendix B. An interface was established between Lingo and Excel to import the data of the parameters and export the data of the decision variables. Lingo solved this example and found an optimal solution in thirty three seconds (00:00:33). The most important output of this stage is the initial plan of production without the sequence. The sequence will be determined in the following stages. Table 21 shows the output of stage one which will be used as an input in the next stage in order to optimize the sequence of the plan while reducing the number of the setups.

Table 21 Stage one output

		T1	T2	T3	T4	T5	T6	T7	T8
A1	g1	0	5298	4214	0	5480	0	322	0
	g2	5260	0	0	0	0	0	834	137
	g3	200	162	0	5480	0	0	0	0
	g4	0	0	1246	0	0	0	0	5323
	g5	0	0	0	0	0	5480	247	0
A2	g1	5480	0	0	0	0	0	0	602
	g2	0	0	2238	810	0	0	0	0
	g3	0	0	3222	155	0	5480	0	4858
	g4	0	5480	0	753	5480	0	0	0
	g5	0	0	0	3702	0	0	5480	0

Table 21 Stage one output (Continued)

		T1	T2	T3	T4	T5	T6	T7	T8
A3	g1	0	0	0	0	0	0	0	5480
	g2	0	5480	0	0	495	0	5480	0
	g3	5480	0	2479	0	4965	0	0	0
	g4	0	0	2700	0	0	5480	0	0
	g5	0	0	261	5480	0	0	0	0
A4	g1	0	0	0	0	0	0	0	0
	g2	1781	0	0	0	4276	2178	0	5465
	g3	0	0	0	0	0	0	0	0
	g4	3679	1339	0	5480	0	0	5480	0
	g5	0	4121	5480	0	1184	0	0	0
A5	g1	0	0	5480	0	0	2861	4715	0
	g2	0	0	0	5480	0	2599	0	0
	g3	0	4526	0	0	3993	0	0	0
	g4	1144	0	0	0	1467	0	745	0
	g5	4316	934	0	0	0	0	0	5037

3.7.2 Stage two sequence reduction and optimization. In this stage, the first step is to optimize the sequence of the initial plan in order to use this sequence in determining the available capacity at each period in each affiliate. Table 22 presents a sample of the initial sequence that is provided by this stage.

Table 22 sample of Initial sequence of affiliate 1

		T1	T2	T3	T4	T5	T6	T7	T8
A1	1 to 1	0	0	0	0	0	0	0	0
	1 to 2	0	0	0	0	0	0	1	0
	1 to 3	0	0	0	0	0	0	0	0
	1 to 4	0	0	1	0	0	0	0	0
	1 to 5	0	0	0	0	1	0	0	0
	2 to 1	0	0	0	0	0	0	1	0
	2 to 2	0	0	0	0	0	0	0	0
	2 to 3	1	0	0	0	0	0	0	0
	2 to 4	0	0	0	0	0	0	0	1
	2 to 5	0	0	0	0	0	0	0	0
	3 to 1	0	1	0	1	0	0	0	0
	3 to 2	0	0	0	0	0	0	0	0
	3 to 3	0	0	0	0	0	0	0	0
	3 to 4	0	0	0	0	0	0	0	0
	3 to 5	0	0	0	0	0	0	0	0
	4 to 1	0	0	0	0	0	0	0	0
	4 to 2	0	0	0	0	0	0	0	0
	4 to 3	0	0	1	0	0	0	0	0
	4 to 4	0	0	0	0	0	0	0	0

Table 22 sample of Initial sequence of affiliate 1 (Continued)

		T1	T2	T3	T4	T5	T6	T7	T8
A1	4 to 5	0	0	0	0	0	0	0	0
	5 to 1	0	0	0	0	0	0	0	0
	5 to 2	0	0	0	0	0	0	1	0
	5 to 3	0	0	0	0	0	0	0	0
	5 to 4	0	0	0	0	0	0	0	0
	5 to 5	0	0	0	0	0	0	0	0

After the initial sequence optimization, the algorithm will start the sequence optimization procedure. In this example, only one sequence was reduced in both options. This is expected because the initial solution by the first stage was almost optimal, therefore the second stage depends on the quality of the initial solution. Table 23 presents the new plan after the sequence reduction stage.

Table 23 Stage two output

		T1	T2	T3	T4	T5	T6	T7	T8
A1	g1	0	5298	4214	0	5480	0	924	0
	g2	5260	0	0	0	0	0	834	137
	g3	200	162	0	5480	0	0	0	0
	g4	0	0	1246	0	0	0	0	5323
	g5	0	0	0	0	0	5480	247	0
A2	g1	5480	0	0	0	0	0	0	0
	g2	0	0	2238	810	0	0	0	0
	g3	0	0	3222	155	0	5480	0	4858
	g4	0	5480	0	753	5480	0	0	0
	g5	0	0	0	3702	0	0	5480	0
A3	g1	0	0	0	0	0	0	0	5480
	g2	0	5480	0	0	495	0	5480	0
	g3	5480	0	2479	0	4965	0	0	0
	g4	0	0	2700	0	0	5480	0	0
	g5	0	0	261	5480	0	0	0	0
A4	g1	0	0	0	0	0	0	0	0
	g2	1781	0	0	0	4276	2178	0	5465
	g3	0	0	0	0	0	0	0	0
	g4	3679	1339	0	5480	0	0	5480	0
	g5	0	4121	5480	0	1184	0	0	0
A5	g1	0	0	5480	0	0	2861	4715	0
	g2	0	0	0	5480	0	2599	0	0
	g3	0	4526	0	0	3993	0	0	0
	g4	1144	0	0	0	1467	0	745	0
	g5	4316	934	0	0	0	0	0	5037

A shift was done in A2 at T8 for g1 to A1 T7 g1. Comparing the initial plan from the first stage with the plan from the second stage, an amount of 602 of grade 2 was shifted from affiliate two at period 8 to affiliate one at period 7. This shift reduces the number of sequences by one.

3.7.3 Third stage inventory reduction. In this stage, the heuristic will try to reduce the cumulative inventory in order to reduce the total holding cost. The movement of the cumulative inventory is done from period 1 to period T. This stage allows the lot splitting in order to reduce the most of the inventory. In other words, lot splitting is allowed in the inventory to take advantage of the small available capacities that do not fit the total amount of the cumulative inventory. Previously, it was mentioned that the heuristic provide four different plans due to the nature of the procedure. Table 24 present one of these production plans and Table 25 illustrate a sample of the corresponding optimal sequence for affiliate 1. This plan was generated using option one in stage two and option one in stage three.

Table 24 Production plan from the heuristic

		T1	T2	T3	T4	T5	T6	T7	T8
A1	g1	0	5332	4214	0	5444	0	904	0
	g2	5244	0	0	0	0	0	834	137
	g3	200	162	0	5480	0	0	0	0
	g4	0	0	1272	0	0	0	0	5323
	g5	0	0	0	0	0	5500	247	0
A2	g1	5446	0	0	0	0	0	0	0
	g2	0	0	2273	810	0	0	0	0
	g3	0	0	3222	155	0	5495	0	4858
	g4	0	5399	0	753	5480	0	0	0
	g5	0	0	0	3673	0	0	5480	0
A3	g1	0	0	0	0	0	0	0	5500
	g2	0	5461	0	0	495	0	5480	0
	g3	5480	0	2479	0	4965	0	0	0
	g4	0	0	2755	0	0	5457	0	0
	g5	0	0	261	5480	0	0	0	0
A4	g1	0	0	0	0	0	0	0	0
	g2	1781	0	0	0	4276	2178	0	5465
	g3	0	0	0	0	0	0	0	0
	g4	3679	1339	0	5480	0	0	5491	0
	g5	0	4121	5480	0	1193	0	0	0
A5	g1	0	0	5480	0	0	2897	4715	0
	g2	0	0	0	5480	0	2599	0	0
	g3	0	4526	0	0	3978	0	0	0
	g4	1144	0	0	0	1454	0	770	0
	g5	4316	934	0	0	0	0	0	5037

Table 25 Sample of optimal sequence of the heuristic output

		T1	T2	T3	T4	T5	T6	T7	T8
A1	1 to 1	0	0	0	0	0	0	0	0
	1 to 2	0	0	0	0	0	0	1	0
	1 to 3	0	0	0	0	0	0	0	0
	1 to 4	0	0	1	0	0	0	0	0
	1 to 5	0	0	0	0	1	0	0	0
	2 to 1	0	0	0	0	0	0	1	0
	2 to 2	0	0	0	0	0	0	0	0
	2 to 3	1	0	0	0	0	0	0	0
	2 to 4	0	0	0	0	0	0	0	1
	2 to 5	0	0	0	0	0	0	0	0
	3 to 1	0	1	0	1	0	0	0	0
	3 to 2	0	0	0	0	0	0	0	0
	3 to 3	0	0	0	0	0	0	0	0
	3 to 4	0	0	0	0	0	0	0	0
	3 to 5	0	0	0	0	0	0	0	0
	4 to 1	0	0	0	0	0	0	0	0
	4 to 2	0	0	0	0	0	0	0	0
	4 to 3	0	0	1	0	0	0	0	0
	4 to 4	0	0	0	0	0	0	0	0
	4 to 5	0	0	0	0	0	0	0	0
	5 to 1	0	0	0	0	0	0	0	0
	5 to 2	0	0	0	0	0	0	1	0
	5 to 3	0	0	0	0	0	0	0	0
	5 to 4	0	0	0	0	0	0	0	0
	5 to 5	0	0	0	0	0	0	0	0

This heuristic was coded using Matlab 15.0, it took seventeen seconds to generate the four outputs. A sample of the code can be found in Appendix C. The final stage is to evaluate the optimal profit of each plan in order to compare it with the exact solution. Table 27 presents the comparison between the exact solution and the heuristic solution.

Table 26 Heuristic Performance

HEURISTIC			
CLP	2nd and 3rd	Plan	Profit
0:00:33	0:00:17	1	169,877,800
		2	169,877,800
		3	169,877,800
		4	169,877,800

It can be seen that the four outcomes have the same profit, which means the four plans are the same. The exact solution computational time is 3:50:20 with a profit of 170,004,400 AED while the heuristic computational time is 00:00:50 with a profit of 169,877,800 AED. Table 27 represents the comparison between the exact and the heuristic solution.

Table 27 Performance of Exact vs Heuristic

Exact		**Heuristic**		
Profit	Time	Profit	Time	Gap
170,004,400	3:50:20	169,877,800	00:00:50	-0.074469%

The gap between the two answers is 0.075%. It is evident that the heuristic performed well in this example in terms of computational time and final results. The efficiency of the heuristic is assessed in Chapter 4 by testing different problems.

Chapter 4: Sensitivity Analysis

In this chapter, sensitivity analysis is performed on the developed mathematical model and the developed heuristic to highlight the difference in computational time with respect to problem size and different cost values as well. The sensitivity will be performed in three different parts: the first part will tackle the changes in the computational time with respect to changes in costs; while the second part will look at the complexity of the problem in terms of size; finally the third part will present the performance of the heuristic when large size problems are solved.

The first step in the sensitivity analysis is to fix a reference case in order to compare it with the other results. The reference case is solved with the following assumptions. The size of the parameters of this case is as follow.

- Number of Suppliers: 4.
- Number of Raw Materials: 4.
- Number of Affiliates: 4.
- Number of Grades: 4.
- Number of Warehouses: 4.
- Number of Customers: 10.
- Number of Periods: 6.

The data that was used in this case is randomly generated, these data is uniformly distributed according to the following limits.

- Transportation cost from supplier *s* to affiliate *a*: U [1, 10] AED.
- Transportation cost from affiliate *a* to warehouse *w*: U [1, 10] AED.
- Transportation cost from warehouse *w* to customer *k*: U [5, 15] AED.
- Purchasing Cost of raw material *r* from supplier *s* to affiliate *a*: U [4, 8] AED.
- Profit of grade *g* in affiliate *a* during time period *t*: U [500, 1000] AED.
- Transitional cost between grade *g* and *h* in affiliate *a* during time period *t*.

These costs were generated using the following procedure:

For simplicity, the absolute difference is considered in generating the transitional matrix i.e. a transition from grade one to three is equal to a transition from grade three to one. If the difference between *g* and *h* is 1, the costs follow U [10, 50] AED, if the difference between *g* and *h* is 2, the costs follow U [50, 100] AED, If the difference between *g* and *h* is 3, the costs follow U [100, 150] AED and if the difference between *g* and *h* is 4, the costs follow U [150, 200] AED.

- Inventory carrying cost of grade *g* in warehouse *w* during time period *t*: U [1, 10] AED.
- Availability of raw material *r* from supplier *s*: Constant number of 10^6.
- Production capacity of affiliate *a* during time period *t*: Constant number of 5500.
- Transitional material between grade *g* and *h* in affiliate *a* during time period *t*.

  These amounts were generated in a special procedure. For simplicity, the absolute difference is considered in generating the transitional matrix i.e. a transition from grade one to three is equal to a transition from grade three to one. If the difference between *g* and *h* is 1, the amount follow U [1, 5], if the difference between *g* and *h* is 2, the amount follow U [5, 10], if the difference between *g* and *h* is 3, the amount follow U [10, 15] and if the difference between *g* and *h* is 4, the amount follow U [15, 20].

- Quantity needed of raw material *r* to produce grade *g* in affiliate *a*: U [0, 5].
- Demand of grade *g* for customer *k* during time period *t*: U [100, 1000].
- Percentage of the excess capacity allocated to grade *g* at each period *t*. it was assumed in this example that no excess capacity should be allocated to any grade, therefore the percentage is zero across all grades and periods.

4.1 Parameters Values Variation

The aim of this part is to study the effect of varying the different costs independently in order to illustrate the effect of these costs on the computational time. In addition to the profits, four different costs were considered in this phase: holding cost, raw material cost, transitional cost, and transportation cost. A total of twenty problems were tested in this part, four problems for each cost.

4.1.1 Holding cost variation. In these trials, the holding costs in the different warehouses were varied independently in order to study the effect of these variations on the total computational time while testing the performance of the heuristic. In the first trial, the holding costs are multiplied by 0.25 factor, in the second trial, the holding costs are multiplied by 0.5 factor, in the third trial, the reference case is considered without any changes to the holding costs, In the fourth trial the holding costs are multiplied by 2 factor and in the fifth trial the holding costs are multiplied by 4 factor. Figure 21 represents the computational time of each trial in the exact solution and the heuristic solution. It can be noticed that the holding costs values affect the CT in the exact solution with a direct variation relationship where the higher the holding costs the greater CT, while in the heuristic the holding costs affect the CT with a no correlation relationship.

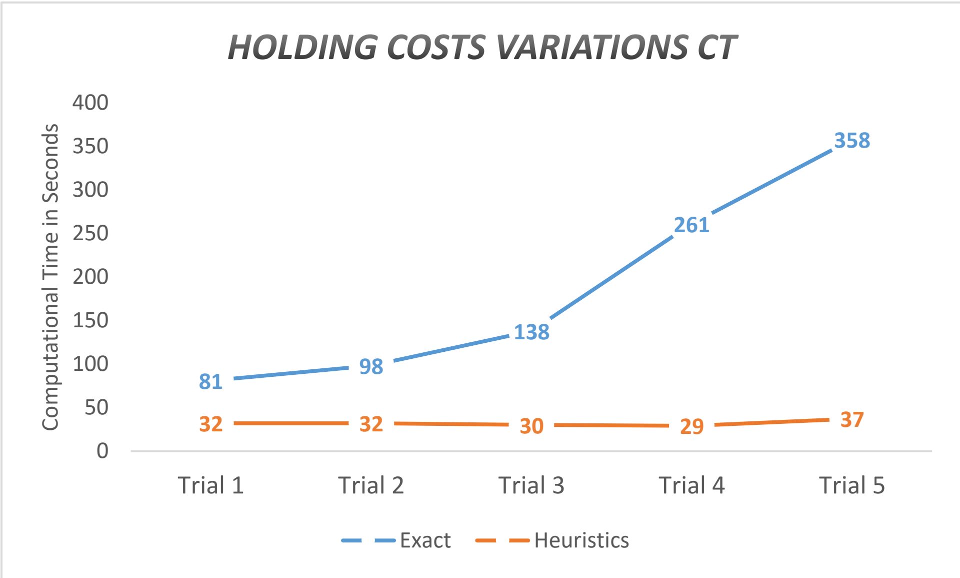

Figure 21 Holding Costs Variations CT

Table 28 represents a comparison between the exact optimal solution and the heuristic solutions when the holding costs were varied. The gap between the heuristic solutions and the exact solutions are varying in each trial, this is expected because the aim of stage three is to reduce the cumulative holding costs and varying these costs will affect the quality of the solution. The average gap in these trials is 0.45%.

Table 28 Holding Costs Variations Heuristic Gap

Trials	**Exact**	**Heuristic**		
	Profit	Plan	Profit	Gap %
Trial 1	89,417,130	1	89,000,040	-0.47%
		2	89,000,040	-0.47%
		3	89,000,040	-0.47%
		4	89,000,040	-0.47%

Table 28 Holding Costs Variations Heuristic Gap (Continued)

Trials	Exact	Heuristic		
	Profit	Plan	Profit	Gap %
Trial 2	89,413,430	1	89,000,640	-0.46%
		2	89,000,640	-0.46%
		3	89,000,640	-0.46%
		4	89,000,640	-0.46%
Trial 3	89,392,620	1	88,992,260	-0.45%
		2	88,992,260	-0.45%
		3	88,992,260	-0.45%
		4	88,992,260	-0.45%
Trial 4	89,378,070	1	88,990,210	-0.43%
		2	88,990,210	-0.43%
		3	88,990,210	-0.43%
		4	88,990,210	-0.43%
Trial 5	89,309,610	1	88,906,020	-0.45%
		2	88,906,020	-0.45%
		3	88,906,020	-0.45%
		4	88,906,020	-0.45%

4.1.2 Profits variation. In these trials, the profits in the different affiliates were varied independently in order to study the effect of these variations on the total computational time while testing the performance of the heuristic. In the first trial, the profits are multiplied by 0.25 factor, in the second trial, the profits are multiplied by 0.5 factor, in the third trial, the reference case is considered without any changes to the profits, in the fourth trial the profits are multiplied by 2 factor and in the fifth trial the profits are multiplied by 4 factor. Figure 22 presents the variation effect on the CT for the exact and the heuristic solution. The computational time in the exact and the heuristic are fluctuating in each trial without a clear relationship. This is expected because the profits in each affiliate affect the assignment of the raw materials from which supplier to which affiliate, the transportation from each affiliate to each warehouse and the sequence of the production plans in each affiliate.

As for the heuristic performance when the profits are varied, Table 29 represents a comparison between the exact optimal solution and the heuristic solutions. The gap between the heuristic solutions and the exact solutions are varying in each trial. This is expected because in the second stage, the shifting criteria depend on the profits of each affiliate, therefore the performance of the heuristic will be affected based on it. The average gap in these trials is 0.61%.

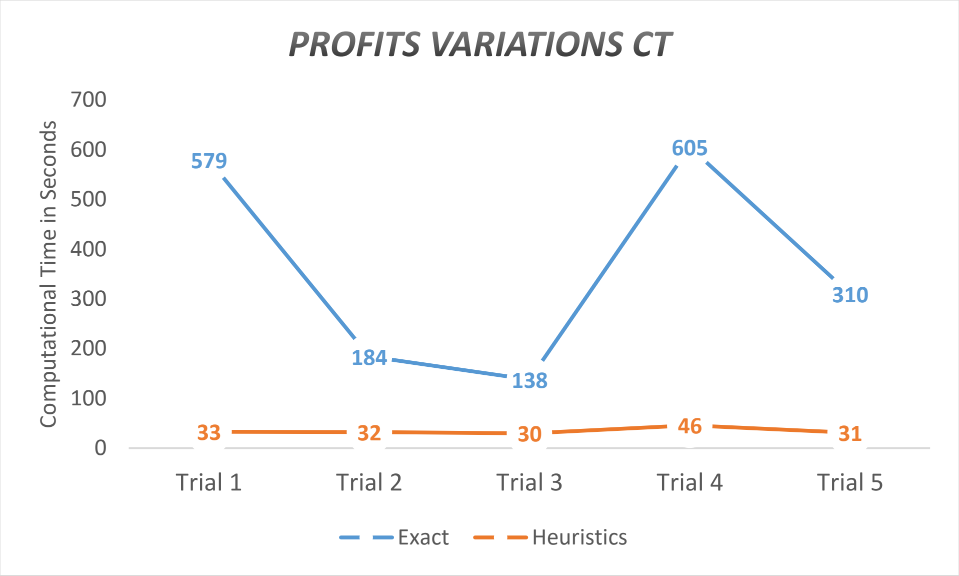

Figure 22 Profits Variations CT

Table 29 Profits Variations Gap

Trials	Exact	Heuristic		
	Profit	Plan	Profit	Gap
Trial 1	15,988,320	1	15,669,750	-1.99%
		2	15,668,760	-2.00%
		3	15,669,750	-1.99%
		4	15,668,760	-2.00%
Trial 2	40,673,500	1	40,636,310	-0.09%
		2	40,636,230	-0.09%
		3	40,636,310	-0.09%
		4	40,636,230	-0.09%
Trial 3	89,392,620	1	88,992,060	-0.45%
		2	88,992,060	-0.45%
		3	88,992,060	-0.45%
		4	88,992,060	-0.45%
Trial 4	195,165,400	1	194,832,500	-0.17%
		2	194,832,500	-0.17%
		3	194,832,500	-0.17%
		4	194,832,500	-0.17%
Trial 5	398,091,000	1	396,710,500	-0.35%
		2	396,710,500	-0.35%
		3	396,710,500	-0.35%
		4	396,710,500	-0.35%

4.1.3 Raw material costs variation. In these trials, the raw materials costs for different suppliers were varied independently in order to study the effect of these variations on the total computational time while testing the performance of the heuristic. In the first trial, the raw materials costs are multiplied by 0.25 factor, in the second trial, the raw materials costs are multiplied by 0.5 factor, in the third trial, the reference case is considered without any changes to the raw materials costs, in the fourth trial the raw materials costs are multiplied by 2 factor and in the fifth trial the raw materials costs are multiplied by 4 factor. Figure 23 presents the variation effect on the CT for the exact and the heuristic solution. It can be noticed that the computational time in the exact and the heuristic solution is affected slightly. This is due to the costs of the raw materials only that affect the choice of which supplier to supply the raw materials. Table 30 represents a comparison between the exact optimal solution and the heuristic solutions when the raw materials costs are varied. The gap between the heuristic solutions and the exact solutions are varying in each trial. As expected, the heuristic ignores the raw materials costs when the shifts from one affiliate to another is done, therefore the costs might be higher because the heuristic forces the affiliate to operate without a consideration of the total raw materials costs for each affiliate. The average gap in these trials is 0.65%.

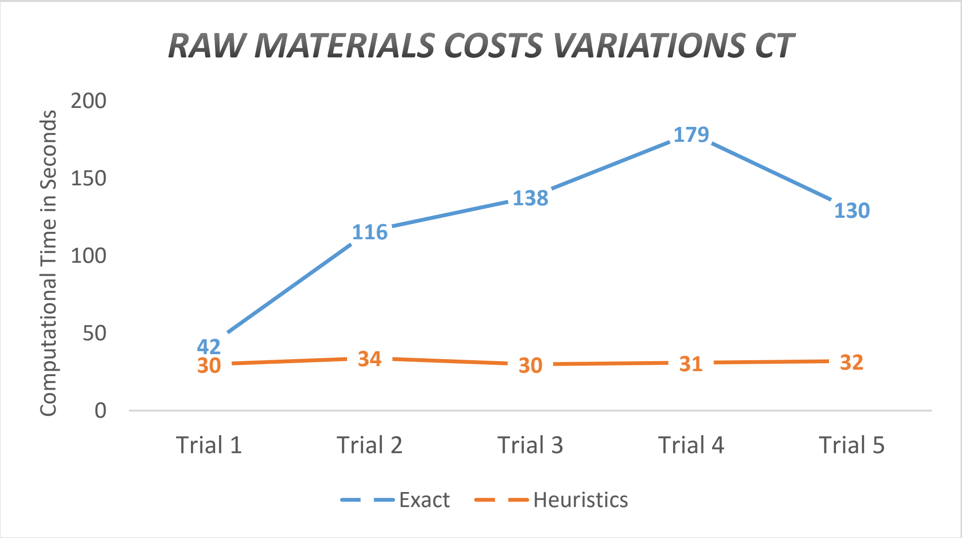

Figure 23 Raw Material Costs Variations CT

Table 30 Raw Material Costs Variations Gap

Trials	Exact	Heuristic		
	Profit	Plan	Profit	Gap
Trial 1	93,799,300	1	93,246,920	-0.59%
		2	93,246,920	-0.59%
		3	93,246,920	-0.59%
		4	93,246,920	-0.59%
Trial 2	92,248,950	1	91,692,290	-0.60%
		2	91,692,290	-0.60%
		3	91,692,290	-0.60%
		4	91,692,290	-0.60%
Trial 3	89,392,620	1	88,992,060	-0.45%
		2	88,992,060	-0.45%
		3	88,992,060	-0.45%
		4	88,992,060	-0.45%
Trial 4	85,054,600	1	84,749,510	-0.36%
		2	84,749,510	-0.36%
		3	84,749,510	-0.36%
		4	84,749,510	-0.36%
Trial 5	77,195,810	1	76,245,370	-1.23%
		2	76,245,370	-1.23%
		3	76,245,370	-1.23%
		4	76,245,370	-1.23%

4.1.4 Transitional costs variation. In these trials, the transitional costs in the different warehouses were varied independently in order to study the effect of these variations on the total computational time while testing the performance of the heuristic. In the first trial, the transitional costs are multiplied by 0.25 factor, in the second trial, the transitional costs are multiplied by 0.5 factor, in the third trial, the reference case is considered without any changes to the transitional costs, in the fourth trial the transitional costs are multiplied by 2 factor and in the fifth trial the transitional costs are multiplied by 4 factor. Figure 24 presents the variation effect on the CT for the exact and the heuristic solution.

The computational time in the exact and the heuristic solution is affected slightly, this is justified because transitional costs across the different affiliates are affected with the same weight. The optimal sequence would not change if the costs change with the same weights. However, the net profit is affected because the transitional costs were changed. As for the heuristic performance when the transitional costs are varied, Table 31 represents a comparison between the exact optimal solution and the heuristic solutions. The gap between the heuristic solutions and the exact solutions has the same value across the different trials. As expected the heuristic will keep providing the same optimal solution in each trial but with different profits. The average gap in these trials is 0.45%.

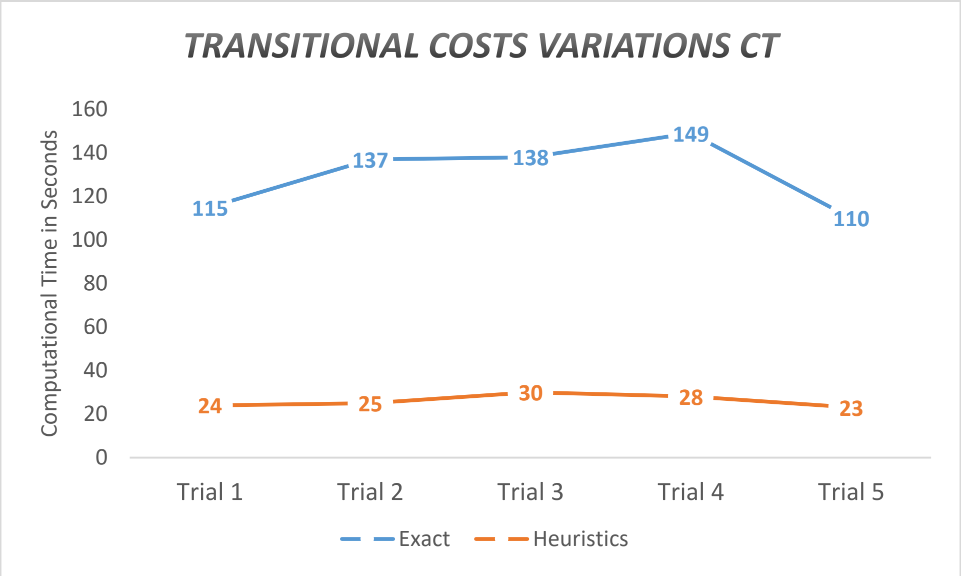

Figure 24 Transitional Costs Variations CT

Table 31 Transitional Costs Variations Gap

Trials	**Exact**	**Heuristic**		
	Profit	Plan	Profit	Gap
Trial 1	89,398,330	1	88,995,860	-0.45%
		2	88,995,860	-0.45%
		3	88,995,860	-0.45%
		4	88,995,860	-0.45%
Trial 2	89,395,300	1	88,994,590	-0.45%
		2	88,994,590	-0.45%
		3	88,994,590	-0.45%
		4	88,994,590	-0.45%
Trial 3	89,392,620	1	88,992,060	-0.45%
		2	88,992,060	-0.45%
		3	88,992,060	-0.45%
		4	88,992,060	-0.45%
Trial 4	89,385,590	1	88,986,990	-0.45%
		2	88,986,990	-0.45%
		3	88,986,990	-0.45%
		4	88,986,990	-0.45%
Trial 5	89,376,500	1	88,976,850	-0.45%
		2	88,976,850	-0.45%
		3	88,976,850	-0.45%
		4	88,976,850	-0.45%

4.1.5 Transportation costs variation. In these trials, the transportations costs in the three different stages (from suppliers to affiliates, affiliates to warehouses and warehouses to customers) were varied independently in order to study the effect of these variations on the total computational time while testing the performance of the heuristic. In the first trial, the transportations costs are multiplied by 0.25 factor, in the second trial, the transportations costs are multiplied by 0.5 factor, in the third trial, the reference case is considered without any changes to the transportations costs, in the fourth trial the transportations costs are multiplied by 2 factor and in the fifth trial the transportations costs are multiplied by 4 factor. Figure 25 presents the variation effect on the CT for the exact and the heuristic solution. The computational time in the exact and the heuristic solution varies greatly. This is expected because the costs of the transportation in the three stages affect the assignment of the production plans in each affiliate. This effect can be compared with the profits variations effect since the relationship between the costs and the computational time is no correlation relationship.

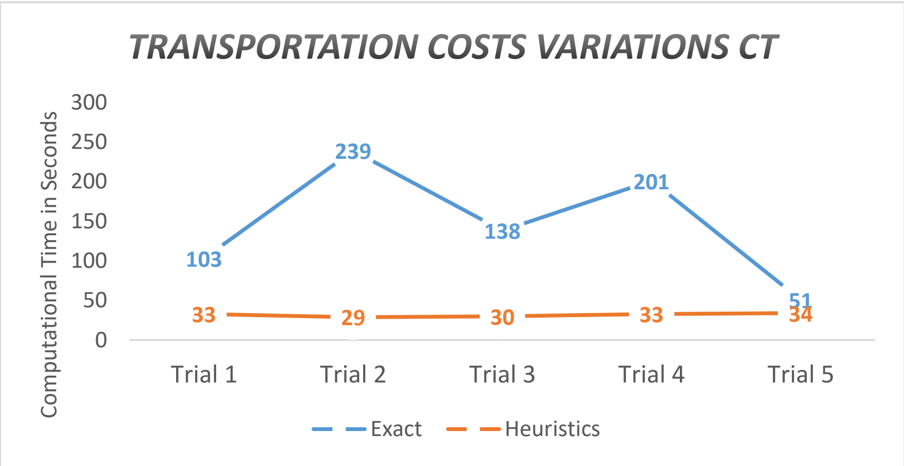

Figure 25 Transportation Costs Variations CT

As for the heuristic performance when the transportation costs are varied, Table 32 represents a comparison between the exact optimal solution and the heuristic solutions. The gap between the heuristic solutions and the exact solutions vary with the different trials. This is justified because the heuristic ignores the transportation costs between the different stages in the supply chain which affect the gap between the exact and heuristic solution. The average gap in these trials is 0.53%.

Table 32 Transportation Costs Variations Gap

Trials	Exact	Heuristic		
	Profit	Plan	Profit	Gap
Trial 1	93,037,690	1	92,670,480	-0.39%
		2	92,670,480	-0.39%
		3	92,670,480	-0.39%
		4	92,670,480	-0.39%
Trial 2	91,173,710	1	90,802,740	-0.41%
		2	90,802,740	-0.41%
		3	90,802,740	-0.41%
		4	90,802,740	-0.41%
Trial 3	89,392,620	1	88,992,060	-0.45%
		2	88,992,060	-0.45%
		3	88,992,060	-0.45%
		4	88,992,060	-0.45%
Trial 4	85,003,000	1	84,577,985	-0.50%
		2	84,577,985	-0.50%
		3	84,577,985	-0.50%
		4	84,577,985	-0.50%
Trial 5	80,018,430	1	79,314,240	-0.88%
		2	79,314,240	-0.88%
		3	79,314,240	-0.88%
		4	79,314,240	-0.88%

4.2 Parameters Size Variation

The aim of this part is to study the effect of varying the size of the parameters independently in order to illustrate the effect of these parameters on the computational time. When the size of the problem increases, the complexity increases. Therefore the larger the size of the parameter the more complex the problem is. In this part, the effects of the most important parameters that can change often were considered. These parameters are:

- Number of Periods.
- Number of Grades.
- Number of Affiliates.

4.2.1 Periods size variations. The most frequently changed parameter is the periods. The size of this parameter depends on the planning horizon which can be in terms of days, weeks or months. Five different problems with different combinations are solved to study the effect of this parameter size on the computational time and the performance of the heuristic. The reference case is considered the first example then the periods are increased by two periods in each example. Table 33 presents the combinations of these examples.

Table 33 Combinations of Different Periods

Example	Parameters									
	Suppliers	Raw Materials	Affiliates	Grades	Warehouses	Customers	Periods	Total Number of Nodes	# of Variables	# of Constraints
P6	4	4	4	4	4	10	**6**	36	4917	2148
P8	4	4	4	4	4	10	**8**	38	6331	2766
P10	4	4	4	4	4	10	**10**	40	7745	3384
P12	4	4	4	4	4	10	**12**	42	9159	4002
P14	4	4	4	4	4	10	**14**	44	10573	4620

In each example, as the number of periods is increased, the total number of the variables and constraints also are increased exponentially. The total computational time of the exact and the heuristic are presented in Table 34.

The exact solution computational time fluctuates with a no correlation relationship, however, the impact of the different sizes of the periods does not affect the heuristic computational time significantly. Figure 26 presents the computational time of the exact and the heuristic solution. Overall, the different sizes of the period parameter do not impact the computational time significantly within the range of 6 to 14 periods. Table 35 presents the performance of the heuristic when different examples with different periods are solved. The gap percentage is between 0.16% and 0.48% which indicates a good quality of the solution. However, the relation between the different sizes of periods and the solution gap is no correlation relationship.

Table 34 Different Periods CT

Example	**Computational Time**	
	Exact	Heuristic
P6	00:02:18	00:00:30
P8	00:00:41	00:00:29
P10	00:01:53	00:00:27
P12	00:01:50	00:00:29
P14	00:03:23	00:00:33

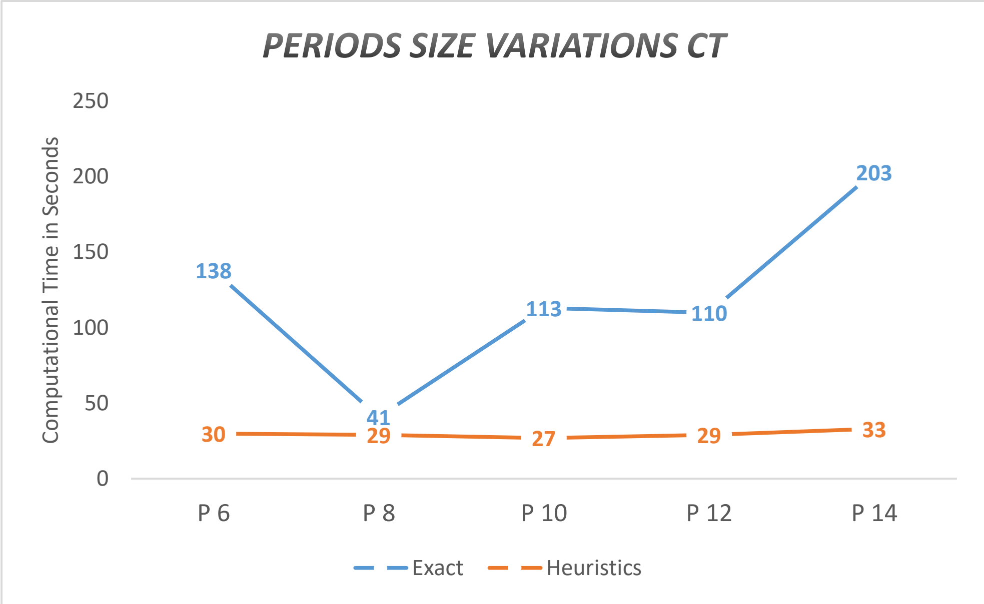

Figure 26 Periods Size Variations CT

Table 35 Periods Size Variations Gap

Examples	**Exact**	**Heuristic**		
	Profit	Plan	Profit	Gap
P 6	89,392,620	1	88,992,060	-0.45%
		2	88,992,060	-0.45%
		3	88,992,060	-0.45%
		4	88,992,060	-0.45%
P 8	130,265,200	1	130,059,000	-0.16%
		2	130,057,600	-0.16%
		3	130,058,800	-0.16%
		4	130,057,600	-0.16%
P 10	163,162,500	1	162,661,200	-0.31%
		2	162,661,200	-0.31%
		3	162,661,200	-0.31%
		4	162,661,200	-0.31%
P 12	193,926,100	1	193,477,300	-0.23%
		2	193,478,500	-0.23%
		3	193,478,500	-0.23%
		4	193,478,500	-0.23%
P 14	224,726,100	1	223,648,500	-0.48%
		2	223,649,800	-0.48%
		3	223,649,800	-0.48%
		4	223,649,800	-0.48%

4.2.2 Number of grades variations.

4.2.2 Number of grades variations. This parameter changes less frequently than the periods. The size of this parameter depends on the number of the different grades that each affiliate is willing to produce. Also, the optimal production plan for each affiliate affects the number of grades to be produced in each period because a different number of grades can be produced in different periods. Five different problems with different combinations are solved to study the effect of this parameter size on the computational time and the performance of the heuristic.

The first example starts with 2 grades, then each example is increased by 1 grade to reach 6 grades. Table 36 presents the combinations of these examples. In each example, as the number of grades increased, the total number of the variables and constraints are increased exponentially. The total computational time of the exact and heuristic are presented in Table 37.

Table 36 Combinations of Different Grades

Example	Parameters									
	Suppliers	Raw Materials	Affiliates	Grades	Warehouses	Customers	Periods	Total Number of Nodes	# of Variables	# of Constraints
G2	4	4	4	**2**	4	10	6	34	2805	882
G3	4	4	4	**3**	4	10	6	35	3833	1431
G4	4	4	4	**4**	4	10	6	36	4917	2148
G5	4	4	4	**5**	4	10	6	37	6057	3033
G6	4	4	4	**6**	4	10	6	38	7253	4086

Table 37 Different Grades CT

Example	Computational Time	
	Exact	Heuristic
G2	00:00:04	00:00:02
G3	00:00:24	00:00:04
G4	00:02:18	00:00:15
G5	00:15:09	00:00:15
G6	**02:00:00**	00:00:23

The exact solution computational time fluctuates with a direct variation relationship, where the computational time increases when the number of grades increases. As for the heuristic computational time, the relationship has a direct variation also where the computational time increases when the size of the grades increase. It must be noted that in the fifth example where the number of grades is 6, the exact solution was not found in a short time, therefore the model was stopped after 02:00:00 hours and the upper bound was considered as the solution of the exact model. Figure 27 presents the computational time of the exact and the heuristic solution. Overall, the different sizes of the grade parameter impact the computational time significantly.

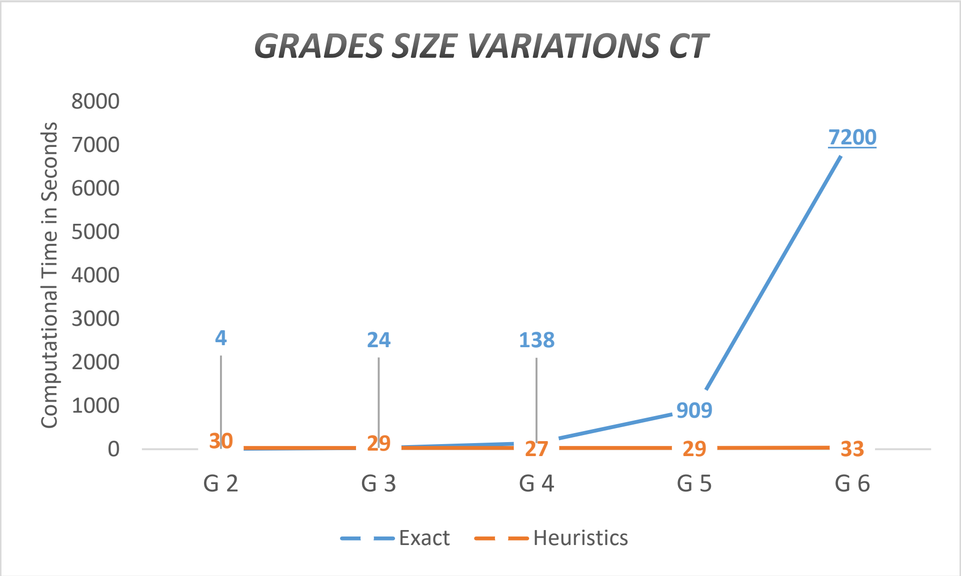

Figure 27 Grades Size Variations CT

The heuristic computational time in the first two examples is greater than the exact computational time. This is justified because the number of the grades are the most critical parameter in the model since it affects the complexity. As mentioned earlier in chapter 3, the sequence-dependent setup costs are the most complex part of the model, therefore when the number of grades increase, the complexity increases exponentially. When the number of grades is small such as 2 and 3, the model becomes simpler which enable the exact model to find the optimal solution faster than the heuristic because the heuristic uses an iterative process that takes more time in the simple problems. Table 38 represents the performance of the heuristic with respect to the different sizes of the grade parameter. The gap percentage is between 0.22% and 0.92% which indicates a good quality of the solution, however, the relation between the different sizes of grades and the solution gap is no correlation relationship.

Table 38 Grades Size Variations Gap

Examples	Exact	Heuristic		
	profit	plan	profit	gap
G 2	50,683,510	1	50,411,190	-0.54%
		2	50,411,190	-0.54%
		3	50,411,190	-0.54%
		4	50,411,190	-0.54%
G 3	72,919,400	1	72,758,880	-0.22%
		2	72,758,880	-0.22%
		3	72,758,880	-0.22%
		4	72,758,880	-0.22%
G 4	89,392,620	1	88,992,060	-0.45%
		2	88,992,060	-0.45%
		3	88,992,060	-0.45%
		4	88,992,060	-0.45%
G 5	117,151,200	1	116,071,800	-0.92%
		2	116,070,800	-0.92%
		3	116,071,400	-0.92%
		4	116,070,800	-0.92%
G 6	143,051,200	1	142,148,200	-0.63%
		2	142,149,600	-0.63%
		3	142,147,000	-0.63%
		4	142,149,600	-0.63%

4.2.3 Affiliates size variations. This parameter changes only when the company changes the joint-ventures either by increasing or decreasing these ventures. The size of this parameter depends on the number of the different affiliates that can produce the different grades. Five different problems with different combinations are solved to study the effect of this parameter size on the computational time and the performance of the heuristic. The reference case is considered the third example where the first example starts with 2 affiliates, then each example is increased by 1 affiliate to reach 6 affiliates. Table 39 presents the combinations of these examples. In each example, the number of grades is increased. At the same time, the total number of the variables and constraints are increased exponentially. The total computational time of the exact and the heuristic are presented in Table 40.

The exact solution computational time fluctuates with a direct variation relationship, where the computational time increases when the number of affiliates increases. As for the heuristic computational time, the relationship has a direct variation also where the computational time increases when the size of the affiliates increase.

Figure 28 presents the computational time of the exact and the heuristic solution. Table 41 presents the performance of the heuristic when different examples with different affiliates are solved. The gap percentage is between 0.17% and 0.86% which indicates a good quality of the solution, however, the relation between the different sizes of periods and the solution gap is no correlation relationship.

Table 39 Combinations of Different Affiliates

Example	Parameters									
	Suppliers	Raw Materials	Affiliates	Grades	Warehouses	Customers	Periods	Total Number of	# of Variables	# of Constraints
A2	4	4	**2**	4	4	10	6	34	3629	1344
A3	4	4	**3**	4	4	10	6	35	4273	1746
A4	4	4	**4**	4	4	10	6	36	4917	2148
A5	4	4	**5**	4	4	10	6	37	5561	2550
A6	4	4	**6**	4	4	10	6	38	6205	2952

Table 40 Different Affiliates CT

Example	Computational Time	
	Exact	Heuristic
G2	00:00:22	00:00:20
G3	00:01:02	00:00:22
G4	00:02:18	00:00:30
G5	00:13:36	00:00:34
G6	00:28:50	00:00:34

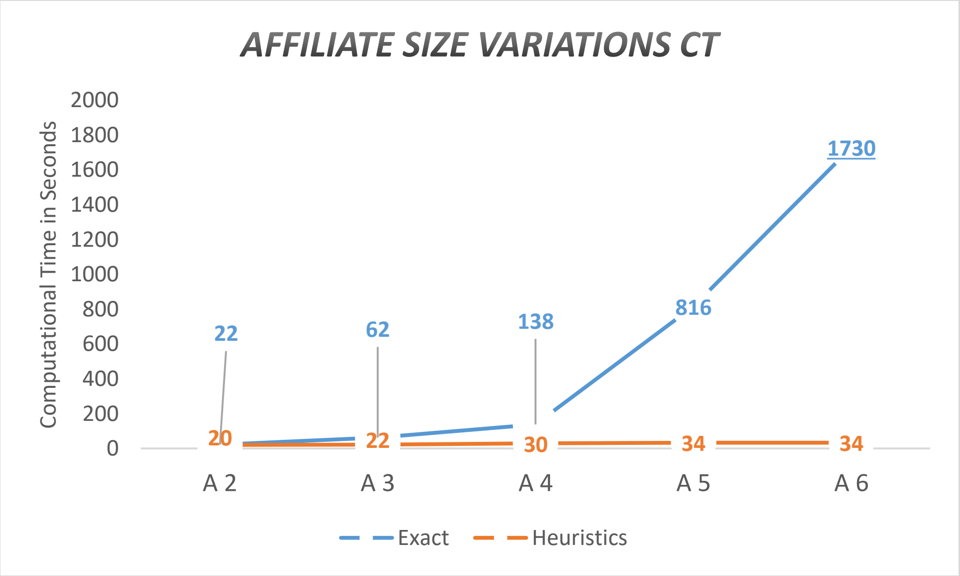

Figure 28 Affiliate Size Variations CT

Table 41 Affiliate Size Variations Gap

Examples	Exact	Heuristic		
	profit	plan	profit	gap
A 2	87,934,190	1	87,181,320	-0.86%
		2	87,176,360	-0.86%
		3	87,181,320	-0.86%
		4	87,176,360	-0.86%
A 3	89,706,630	1	89,556,040	-0.17%
		2	89,556,040	-0.17%
		3	89,556,040	-0.17%
		4	89,556,040	-0.17%
A 4	89,392,620	1	88,992,060	-0.45%
		2	88,992,060	-0.45%
		3	88,992,060	-0.45%
		4	88,992,060	-0.45%
A 5	88,933,130	1	88,652,070	-0.32%
		2	88,652,070	-0.32%
		3	88,652,070	-0.32%
		4	88,652,070	-0.32%
A 6	86,552,820	1	86,233,860	-0.37%
		2	86,232,440	-0.37%
		3	86,234,070	-0.37%
		4	86,232,650	-0.37%

4.3 Heuristic Performance in Large Instance Problems

In this section, the performance of the heuristic is tested by solving a relatively large instance problems in comparison with the sizes that were tested in the previous section. The most important criteria in this section is the computational time mainly because the exact model needs a large computational time that can reach days to get a feasible solution and in some cases the exact model can't find a solution, therefore what will be reported here is the computational time. Table 42 represents the combinations that were used in this section to study the effect of the size of the problems on the computational time in the heuristic, in addition to the computational time that was taken in order to solve these problems using the heuristic.

Four different problems were solved with different sizes. The computational time is increasing and has a direct variation relationship with the size of the problem. In the fourth problem, the computational time increased exponentially in comparison with the third example. This is expected since it was proved by Florian et al. [48] that lot sizing by itself is NP-Hard problem, which means when the size increases the complexity also increases. Therefore the computational time increases. Figure 29 represents the computational time of these different examples.

Table 42 large instance problems combinations

Example	Parameters										Heuristic
	Suppliers	Raw Materials	Affiliates	Grades	Warehouses	Customers	Periods	Total Number of Nodes	# of Variables	# of Constraints	Time
E 6	6	6	6	6	6	6	6	42	7257	1807	00:00:26
E 8	8	8	8	8	8	8	8	56	21211	4105	00:01:55
E 10	10	10	10	10	10	10	10	70	49333	7811	00:10:35
E 16	16	16	16	16	16	16	16	112	299827	30737	04:51:42

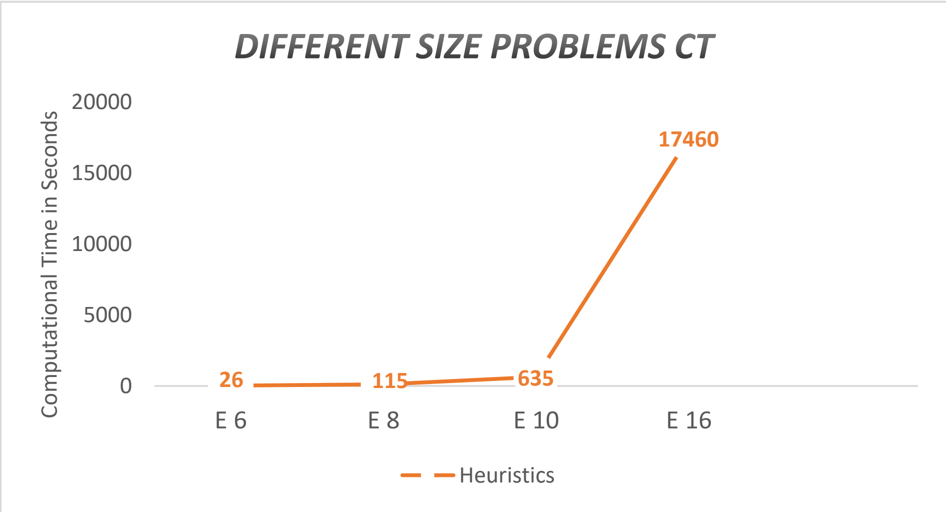

Figure 29 Different Size Problems CT

As a summary, it can be concluded that when it comes to parameters values, the order in terms of the most sensitive to the least sensitive is Profits, Holding Costs, Transportation Costs, Raw Material Costs and Transitional Costs. As for the most sensitive parameters in terms of size, the order in terms of the most sensitive to the least sensitive is Grades, Affiliates and Periods. Finally, the heuristic performs well in the large instance problems in terms of computational time by comparing it with the exact model computational time.

Chapter 5: Conclusion and Future Research

5.1 Conclusion

This research addressed the problem of lot-sizing in production planning domain. Lot-sizing is about the decision of how many orders or lots should be produced in order to satisfy the customer demand while considering a variety of costs such as holding costs and setup costs in order to minimize the total cost. Lot-sizing has different characteristics and properties, the well-known characteristics are number of manufacturing levels, capacity constraints, setup time, planning horizon, demand type and inventory shortages. Lot-sizing problems have a wide range of varieties and can be classified into five different categories. These categories are economic lot-sizing problem, discrete lot-sizing problem, proportional lot-sizing problem, general lot-sizing problem and capacitated lot-sizing problem. This research focuses on the capacitated lot-sizing problem that is considered a large bucket problem, where several products can be produced per time period. Capacitated lot-sizing and scheduling problem can be a complex problem when the size increases, hence it is considered as NP-Hard problem. NP-Hard problems cannot be solved in a reasonable time using the commercial optimization software, therefore solution approaches were developed to solve the CLSP in a reasonable time with a good quality solution. The main approaches that are used to solve lot-sizing problems are Mathematical Heuristic, Lagrangean Relaxation, Decomposition and Aggregation, Metaheuristics and Problem-specific Greedy.

The Petrochemical Market is becoming more competitive recently, this is due to the regulations that are introduced by the World Trade Organization to limit the effect on the environment. Therefore companies started to implement different methods such as employing Enterprise-Resource-Planning systems, supply-chain management and optimization tools such as Linear and Integer Programming in order to increase the efficiency and effectiveness of their production systems [11].

The problem under study is concerned with multi-period planning and scheduling for a company that runs multiple affiliates to produce multiple grades for a variety of products. The company owns multiple warehouses and the raw material needed for production is provided by different suppliers. There is a setup cost for switching the production between the different grades and the setup cost is sequence dependent.

In this work, a contribution is represented to the literature to fill the gap that was clearly demonstrated in the literature chapter regarding the integration between the decisions of scheduling and planning of lot-sizing with the logistics functions of transportation and warehousing. The developed model answers questions regarding the amount of each raw material to be purchased from each supplier, sequence of production plans, inventory levels and warehouse selection to satisfy customer's orders. In addition to that, Three Stage Heuristic is developed to solve large instance problems in a reasonable computational time with good quality of answers.

The model was coded using Lingo 15.0 and tested to illustrate its functionality. After that, the heuristics were coded using Matlab 15.0 in order to present a comparison between an exact solution and the heuristics solution in terms of computational time and solution quality.

Finally, a sensitivity analysis was performed in order to study the effect of different parameters in terms of values and size on the developed model and the performance of the developed heuristic. The sensitivity is conducted in three different areas. In the first area, the values of the important parameters are changed to study the effect on the developed model computational time and the heuristic. These parameters are holding costs, profits, raw material costs, transitional costs and transportation costs. In the second area, the size of the important parameters is changed to study the effect on the developed model computational time and the heuristic. These parameters are number of periods, number of grades and number of affiliates. In the third area, the heuristic was tested on large instance problems to study the effect of large sized problems on the computational time. Overall through 40 different problems with different values and sizes, the heuristic performed well where the solution gap was less than 1% in most cases. As a summary from area one and area two, it is concluded that the values of profits, holding costs and transportation costs affect the computational time the most, while the size of the grades and the affiliates effect the computational time the most. As for area three, the heuristic performs well in the large instance problems in terms of computational time by comparing it with the exact model computational time.

5.2 Future Research

The mathematical model that is presented in this work was developed under certain assumptions as mentioned in Chapter 3. For future research, these assumptions can be changed towards more general model such as allowing backlogging, considering lead times of raw material and finished goods transportation, considering delivery dates for customers, adding the flexibility of delivering the finished goods directly from the affiliates to the customers without going through the warehouses, introducing the capacities of the transportation methods used for each stage, introducing multi-level manufacturing products and introducing safety stocks in each warehouse. In addition to that, the integration between the lot-sizing problem and the logistics function can be done for different classes of lot-sizing problems such as Discrete Lot-sizing problem and General Lot-sizing problem.

As for the developed heuristic, the algorithms can be improved by considering the costs in the different stages of the supply chain such as the transportation costs between different stages and the costs of raw materials in each supplier. The functions of the heuristic can be linked with these costs in order to improve the solution of the heuristic. Also, the proposed algorithm can be applied to different fields such as project management scheduling where the heuristic functionality depends on the utilization of the capacity in each period. This field is promising for a future research.

References

[1] Z. Lu, Y. Zhang and X. Han, "Integrating run-based preventive maintenance into the capacitated lot sizing problem," *International Journal of Production Research,* vol. 51, no. 5, pp. 1379–1391, 2013.

[2] B. Almada-Lobo and R. J. James, "Neighbourhood search meta-heuristics for capacitated lot-sizing with," *International Journal of Production Research,* vol. 48, no. 3, pp. 861–878, 2010.

[3] G. H. Goren, S. Tunali and R. Jans, "A review of applications of genetic algorithms in lot sizing," *Journal of Intelligent Manufacturing,* vol. 21, no. 4, pp. 575–590, 2010.

[4] C. H. Glock, E. H. Grosse and J. M. Ries, "The lot sizing problem: A tertiary study," *International Journal of Production Economics,* vol. 155, no. 4, pp. 39-51, 2014.

[5] T. Liu, "Economic lot sizing problem with inventory bounds," *European Journal of Operational Research,* vol. 185, no. 1, pp. 204–215, 2008.

[6] C. Gicquel and M. Minoux, "Multi-product valid inequalities for the discrete lot-sizing and scheduling problem," *Computers & Operations Research,* vol. 54, no. 10, pp. 12-20, 2015.

[7] W. Kaczmarczyk, "Proportional lot-sizing and scheduling problem with identical," *International Journal of Production Research,* vol. 49, no. 9, pp. 2605–2623, 2011.

[8] A. A. Menezes, A. Clark and B. Almada-Lobo, "Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups," *Journal of Scheduling,* vol. 14, no. 2, pp. 209–219, 2011.

[9] G. R. Bitran and H. H. Yanasse, "Computational complexity of the capacitated lot size problem," *Management Science,* vol. 28, no. 10, pp. 1174-1186, 1982.

[10] L. Buschkühl, F. Sahling, S. Helber and H. Tempelmeier, "Dynamic capacitated lot-sizing problems: a classification and review of solution approaches," *OR Spectrum,* vol. 32, no. 2, pp. 231–261, 2010.

[11] H. K. Alfares, "Optimum multi-plant, multi-supplier production planning for multi-grade petrochemicals," *Engineering Optimization,* vol. 41, no. 5, pp. 1–10, 2009.

[12] R. J. James and B. Almada-Lobo, "Single and parallel machine capacitated lot sizing and scheduling: New iterative MIP-based neighborhood search heuristics," *Computers & Operations Research,* vol. 38, no. 12, pp. 1816 - 1825, 2011.

[13] I.-S. Kwak and I.-J. Jeong, "A hierarchical approach for the capacitated lot-sizing and scheduling problem with a special structure of sequence-dependent setups," *International Journal of Production Research,* vol. 49, no. 24, pp. 7425–7439, 2011.

[14] M. Mirabi, "A hybrid simulated annealing for the single-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times and costs and dynamic release of jobs," *International Journal of Advanced Manufacturing Technology,* vol. 54, no. 9-12, pp. 1109–1119, 2011.

[15] I.-S. Shim, H.-C. Kim, H.-H. Doh and D.-H. Lee, "A two-stage heuristic for single machine capacitated lot-sizing and scheduling with sequence-dependent setup costs," *Computers & Industrial Engineering,* vol. 61, no. 4, pp. 920-929, 2011.

[16] C. Sung and C. T. Maravelias, "A mixed-integer programming formulation for the general capacitated lot-sizing problem," *Computers and Chemical Engineering,* vol. 32, no. 1, pp. 244–259, 2008.

[17] G. A. Süer, F. Badurdeen and N. Dissanayake, "Capacitated lot sizing by using multi-chromosome crossover strategy," *Journal of Intelligent Manufacturing,* vol. 19, no. 3, pp. 273–282, 2008.

[18] B. Almada-Lobo, D. Klabjan, M. A. Carravilla and J. F. Oliveira, "Single machine multi-product capacitated lot sizing with sequence-dependent setups," *International Journal of Production Research,* vol. 45, no. 20, pp. 4873–4894, 2007.

[19] C. Almeder, "A hybrid optimization approach for multi-level capacitated lot-sizing problems," *European Journal of Operational Research,* vol. 200, no. 2, pp. 599–606, 2010.

[20] M. Sambasivan and S. Yahya, "ALagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers," *Computers & Operations Research,* vol. 32, no. 3, pp. 537–555, 2005.

[21] J. Xiao, H. Yang, C. Zhang, L. Zheng and J. N. Gupta, "A hybridLagrangian-simulated annealing-based heuristic for the parallel machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times," *Computers & Operations Research,* vol. 63, no. 9, pp. 72-82, 2015.

[22] D. J. Fiorotto and S. A. de Araujo, "Reformulation and a Lagrangian heuristic for lot sizing problem on parallel machines," *Annals of Operations Research,* vol. 217, no. 1, pp. 213-231, 2014.

[23] D. Quadt and H. Kuhn, "Capacitated lot-sizing with extensions: a review," *4OR,* vol. 6, no. 1, pp. 61-83, 2008.

[24] F. M. B. Toledo and V. A. Armentano, "A Lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines," *European Journal of Operational Research,* vol. 175, no. 2, pp. 1070–1083, 2006.

[25] G. R. Mateus, M. G. Ravetti, M. C. de Souza and T. M. Valeriano, "Capacitated lot sizing and sequence dependent setup scheduling: an iterative approach for integration," *Journal of Scheduling*, vol. 13, no. 3, pp. 245–259, 2010.

[26] F. Marinelli, M. E. Nenni and A. Sforza, "Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company," *Annals of Operations Research,* vol. 150, no. 1, pp. 177–192, 2007.

[27] M. Mohammadi, S. M. F. Ghomi, B. Karim and S. A. Torabi, "MIP-based heuristics for lot sizing in capacitated pure flow shop with sequence-dependent setups," *International Journal of Production Research,* vol. 48, no. 10, pp. 2957–2973, 2010.

[28] M. Babaei, M. Mohammadi and S. M. T. F. Ghomi, "A genetic algorithm for the simultaneous lot sizing and scheduling problem in capacitated flow shop with complex setups and backlogging," *International Journal of Advanced Manufacturing Technology,* vol. 70, no. 1-4, pp. 125-134, 2014.

[29] R. Ramezanian, M. Saidi-Mehrabad and E. Teimoury, "A mathematical model for integrating lot-sizing and scheduling problem in capacitated flow shop environments," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 1-4, pp. 347–361, 2013.

[30] M. Mohammadi and S. M. F. Ghomi, "Genetic algorithm-based heuristic for capacitated lot sizing problem in flow shops with sequence-dependent setups," *Expert Systems with Applications,* vol. 38, no. 6, pp. 7201-7207, 2011.

[31] M. Mohammadi, S. A. Torabi, S. M. T. F. Ghomi and B. Karimi, "A new algorithmic approach for capacitated lot-sizing problem in flow shops with sequence-dependent setups," *The International Journal of Advanced Manufacturing Technology,* vol. 49, no. 1-4, pp. 201–211, 2010.

[32] B. Almada-Lobo, D. Klabjan, M. A. Carravilla and J. F. Oliveira, "Multiple machine continuous setup lot sizing with sequence-dependent setups," *Computational Optimization and Applications,* vol. 47, no. 3, pp. 529-552, 2010.

[33] M. C. Nascimento, M. G. Resende and F. M. Toledo, "GRASP heuristic with path-relinking for the multi-plant capacitated lot sizing problem," *European Journal of Operational Research,* vol. 200, no. 3, pp. 747-754, 2010.

[34] M. Zhang, "Capacitated lot-sizing problem with outsourcing," *Operations Research Letters,* vol. 43, no. 5, pp. 479-483, 2015.

[35] C. Almeder, D. Klabjan, R. Traxler and B. Almada-Lobo, "Lead time considerations for the multi-level capacitated lot-sizing problem," *European Journal of Operational Research,* vol. 241, no. 3, pp. 727-738, 2015.

[36] H. Tempelmeier and T. Hilger, "Linear programming models for a stochastic dynamic capacitated lot sizing problem," *Computers & Operations Research,* vol. 59, no. 9, pp. 119-125, 2015.

[37] A. B. Kantas, H. I. Cobuloglu and I. E. Büyüktahtakin, "Multi-source capacitated lot-sizing for economically viable and clean biofuel production," *Journal of Cleaner Production,* vol. 94, no. 5, pp. 116-129, 2015.

[38] Z. Lu, Y. Zhang and X. Han, "Integrating run-based preventive maintenance into the capacitated lot sizing problem with reliability constraint," *International Journal of Production Research,* vol. 51, no. 5, pp. 1379-1391, 2013.

[39] J. Xiao, C. Zhang, L. Zheng and J. N. Gupta, "MIP-based fix-and-optimise algorithms for the parallel machine capacitated lot-sizing and scheduling problem," *International Journal of Production Research,* vol. 51, no. 16, pp. 5011–5028, 2013.

[40] Z. Pan, J. Tang and O. Liu, "Capacitated dynamic lot sizing problems in closed-loop supply chain," *European Journal of Operational Research,* vol. 198, no. 3, pp. 810-821, 2009.

[41] A. Akbalik and B. Penz, "Exact methods for single-item capacitated lot sizing problem with alternative machines and piece-wise linear production costs," *International Journal of Production Economics,* vol. 119, no. 2, pp. 367-379, 2009.

[42] K. K. Haugen, A. Olstad and B. I. Pettersen, "The profit maximizing capacitated lot-size (PCLSP) problem," *European Journal of Operational Research,* vol. 176, no. 1, pp. 165-176, 2007.

[43] H. K. Alfares., "A Mathematical model for optimum petrochemical multi-grade selection, production, and sequencing," *Analysis of Manufacturing Systems,* 2007.

[44] M. Z. Alqahtani, S. Arifusalam and M. M. Ndiaye, "Focused plant optimization strategy for polyethylene multi-grades and multi-sites production" unpublished.

[45] D. Gupta and T. Magnusson, "The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times," *Computers & Operations Research,* vol. 32, no. 4, pp. 727-747, 2005.

[46] B. Almada-Lobo, J. F. Oliveira and C. A. Maria, "A note on "the capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times," *Computers & Operations Research,* vol. 35, no. 4, pp. 1374, 2008.

[47] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, New York: Wiley, 1988.

[48] M. Florian, J. K. Lenstra and A. H. G. Rinnooy Kan, "Deterministic Production Planning: Algorithms and Complexity," *Management Science,* vol. 26, no. 6, pp. 669-679, 1980.

[49] B. J. Coleman, "Technical note: A simple model for optimizing the single," *Production and Operation Management,* vol. 1, no. 2, pp. 225-228, 1992.

Appendix A

Exact Model Lingo Code

```
MODEL:

SETS:

supplier / S1 .. S4 / ; !Index s;

rawmaterial / r1 .. r4 / ; !Index r;

affiliate / a1 .. a5 / ; !Index a;

grade / g1 .. g5 / ; !Index g;

warehouse / w1 .. w4 / ; !Index w;

customer / c1 .. c10 / ; !Index k;

period / t1 .. t9 / : EX , Q , EA; !Index t;

Link1 ( supplier , affiliate , rawmaterial ) : Cost1 , Cost4 ; !Index
(s,a,r);

Link2 ( affiliate , warehouse , grade ) : Cost2 ; !Index (a,w,g);

Link3 ( warehouse , customer , grade ) : Cost3 ; !Index (w,k,g);

Link4 ( affiliate , grade , period ) : Profit1 , X , alpha , V ;
!Index (a,g,t);

Link5 ( affiliate , grade , grade , period ) : Cost5 , O , Z ; !Index
(a,g,h,t);

Link6 ( warehouse , grade , period ) : Cost6 , I , Intial; !Index
(w,g,t);

Link7 ( supplier , rawmaterial ) : capa1 ; !Index (s,r);

Link8 ( affiliate , period ) : capa2 ; !Index (a,t);

Link9 ( affiliate , rawmaterial , grade ) : Beta ; !Index (a,r,g);

Link10 ( grade , customer , period ) : D ; !Index (g,k,t);

Link11 ( supplier , affiliate , rawmaterial , period ) : F1 , F1F;
!Index (s,a,r,t);

Link12 ( affiliate , warehouse , grade , period ) : F2 , F2F; !Index
(a,w,g,t);

Link13 ( warehouse , customer , grade , period ) : F3 , F3F; !Index
(w,k,g,t);

Link14 ( grade , period ) : par ;
ENDSETS
!Objective Function;
```

```
Max = -@sum ( Link11 (s,a,r,t) : Cost1(s,a,r)  * F1(s,a,r,t) ) - @sum
( Link12 (a,w,g,t) : Cost2(a,w,g) * F2(a,w,g,t) )
- @sum ( Link13 (w,k,g,t) : Cost3(w,k,g) * F3(w,k,g,t) ) - @sum (
Link11 (s,a,r,t) : Cost4(s,a,r)  * F1(s,a,r,t) )
+ @sum ( Link4 (a,g,t) : Profit1(a,g,t) * x(a,g,t) ) - @sum ( Link5
(a,g,h,t) : Cost5(a,g,h,t) * O(a,g,h,t) * Z(a,g,h,t) )
- @sum ( Link6 (w,g,t) : Cost6(w,g,t) * I(w,g,t) );

!Supplier Constraints;

@for ( affiliate (a) :
      @for ( rawmaterial (r) :
            @for ( period (t) :
                  @sum ( grade (g) : Beta(a,r,g) * X(a,g,t) ) = @sum
( Link11 (s,a,r,t) : F1(s,a,r,t) ) ) ) );

@for ( supplier (s) :
      @for ( rawmaterial (r) :
            @for ( period (t) :
                  @sum ( Link11 (s,a,r,t) : F1(s,a,r,t) ) <=
capa1(s,r) ) ) );

!Affiliate Constraints;

@for ( affiliate (a) :
      @for ( period (t) :
            @sum ( Link4 (a,g,t) : x(a,g,t) ) + @sum ( Link5
(a,g,h,t) | h #NE# g : O(a,g,h,t) * Z(a,g,h,t) ) <= capa2(a,t) ) );

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( period (t) :
                  X(a,g,t) <= capa2(a,t) * ( @sum ( Link5 (a,h,g,t) |
h#NE#g : Z(a,h,g,t) ) + alpha(a,g,t) ) ) ) );

@for ( affiliate (a) :
      @for ( period (t) :
            @sum ( Link4 (a,g,t) : alpha(a,g,t) ) = 1 ) );

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( period (t) | t #LT# timesize:
                  alpha(a,g,t) + @sum ( Link5 (a,h,g,t) | h#NE#g :
Z(a,h,g,t) ) = alpha(a,g,t+1) + @sum ( Link5 (a,g,h,t) | h#NE#g :
Z(a,g,h,t) ) ) ) );

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( grade (h) | h #NE# g:
                  @for ( period (t) :
                        V(a,g,t) + (gradesize*Z(a,g,h,t)) -
(gradesize-1) - (gradesize*alpha(a,g,t)) <= V(a,h,t) ) ) ) );



@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( period (t) :
                  @sum ( Link5 (a,h,g,t) | h#NE#g : Z(a,h,g,t) ) +
alpha(a,g,t) <=1 ) ) );
```

```
@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( grade (h) | h #NE# g:
                  @for ( period (t) :
                        z(a,g,h,t) <= Bignumber * x(a,h,t) ) ) ) );

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( grade (h) | h #NE# g:
                  @for ( period (t) :
                        200*z(a,g,h,t) <= x(a,h,t) ) ) ) );

@for ( Link4 (a,g,t) : @GIN( x(a,g,t) ) );

timesize = @size ( period );

gradesize = @size ( grade );

!Warehouse Constraints;

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( period (t) :
                  X(a,g,t) = @sum ( Link12 (a,w,g,t) : F2(a,w,g,t) )
) ) );

@for ( warehouse (w) :
      @for ( grade (g) :
            @for ( period (t) | t #EQ# 1:
                  Intial(w,g,t) + @sum ( Link12 (a,w,g,t) :
F2(a,w,g,t) ) = I(w,g,t) + @sum ( Link13 (w,k,g,t) : F3(w,k,g,t) ) )
) );

@for ( warehouse (w) :
      @for ( grade (g) :
            @for ( period (t) | t #GT# 1 :
                  I(w,g,t-1) + @sum ( Link12 (a,w,g,t) : F2(a,w,g,t)
) = I(w,g,t) + @sum ( Link13 (w,k,g,t) : F3(w,k,g,t) ) ) ) );

!Customer Constraints;

@for ( grade (g) :
      @for ( customer (k) :
            @for ( period (t) :
                  @sum ( Link13 (w,k,g,t) : F3(w,k,g,t) ) >= D(g,k,t)
) ) );

@for ( grade (g) :
      @for ( period (t) :
            @sum ( Link13 (w,k,g,t) : F3(w,k,g,t) ) = @sum ( Link10
(g,k,t) : D(g,k,t) ) + @sum ( Link14 (g,t) : par(g,t) * EX(t) ) ));

!Binary Constraints;


@for ( Link5 (a,h,g,t) : @BIN( Z(a,h,g,t) ) );

@for ( Link4 (a,g,t) : @BIN( alpha(a,g,t) ) );
```

```
@for ( Link6 (w,g,t) | t #GT# 7  : I(w,g,t) = 0 );

!Excess;

@for ( Period (t) :
      EX(t) = @sum ( Link8 (a,t) : capa2(a,t) ) - @sum(Link4 (a,g,t)
:X(a,g,t)) - @sum (Link5 (a,g,h,t) : z(a,g,h,t) * O(a,g,h,t)  ));

@for ( Link11 (s,a,r,t) : F1F(s,a,r,t) = F1(s,a,r,t));
@for ( Link12 (a,w,g,t) : F2F(a,w,g,t) = F2(a,w,g,t));
@for ( Link13 (w,k,g,t) : F3F(w,k,g,t) = F3(w,k,g,t));
DATA:

Bignumber = 1000000;

Cost1, Cost2, Cost3, Cost4, Profit1, Cost5, Cost6, capa1, capa2, O,
Beta, D, Intial, par =
@OLE ( 'C:\Users\sari\Desktop\Sensitivity analysis\main
example\exact\main.xlsx' , 'Cost1' , 'Cost2' , 'Cost3' , 'Cost4' ,
'Profit1'
, 'Cost5' , 'Cost6' , 'capa1' , 'capa2' , 'O' , 'Beta' , 'D' ,
'Intial' , 'par' );

@OLE (  'C:\Users\sari\Desktop\Sensitivity analysis\main
example\exact\main.xlsx' , 'x' , 'z' , 'I' , 'f1f' , 'f2f' , 'f3f' )
= x, z , I, f1f , f2f , f3f ;


ENDDATA
```

Appendix B

Heuristics First Stage Model Lingo Code

```
MODEL:

SETS:

supplier / S1 .. S4 / ; !Index s;

rawmaterial / r1 .. r4 / ; !Index r;

affiliate / a1 .. a5 / ; !Index a;

grade / g1 .. g5 / ; !Index g;

warehouse / w1 .. w4 / ; !Index w;

customer / c1 .. c10 / ; !Index k;

period / t1 .. t9 / : EX , Q , EA; !Index t;

Link1 ( supplier , affiliate , rawmaterial ) : Cost1 , Cost4 ; !Index
(s,a,r);

Link2 ( affiliate , warehouse , grade ) : Cost2 ; !Index (a,w,g);

Link3 ( warehouse , customer , grade ) : Cost3 ; !Index (w,k,g);

Link4 ( affiliate , grade , period ) : Profit1 , X , alpha , V , Y ,
Cost7 , B ; !Index (a,g,t);

Link5 ( affiliate , grade , grade , period ) : Cost5 , O , Z ; !Index
(a,g,h,t);

Link6 ( warehouse , grade , period ) : Cost6 , I , Intial; !Index
(w,g,t);

Link7 ( supplier , rawmaterial ) : capa1 ; !Index (s,r);

Link8 ( affiliate , period ) : capa2 ; !Index (a,t);

Link9 ( affiliate , rawmaterial , grade ) : Beta ; !Index (a,r,g);

Link10 ( grade , customer , period ) : D ; !Index (g,k,t);

Link11 ( supplier , affiliate , rawmaterial , period ) : F1 ; !Index
(s,a,r,t);

Link12 ( affiliate , warehouse , grade , period ) : F2 ; !Index
(a,w,g,t);

Link13 ( warehouse , customer , grade , period ) : F3 ; !Index
(w,k,g,t);

Link14 ( grade , period ) : par ;
ENDSETS
Max = -@sum ( Link11 (s,a,r,t) : Cost1(s,a,r)  * F1(s,a,r,t) ) - @sum
( Link12 (a,w,g,t) : Cost2(a,w,g) * F2(a,w,g,t) )
```

```
- @sum ( Link13 (w,k,g,t) : Cost3(w,k,g) * F3(w,k,g,t) ) - @sum (
Link11 (s,a,r,t) : Cost4(s,a,r)  * F1(s,a,r,t) )
+ @sum ( Link4 (a,g,t) : Profit1(a,g,t) * x(a,g,t) )
- @sum ( Link4 (a,g,t) : Cost7(a,g,t) * Y(a,g,t))
- @sum ( Link6 (w,g,t) : Cost6(w,g,t) * I(w,g,t) );

!Supplier Constraints;

@for ( affiliate (a) :
      @for ( rawmaterial (r) :
            @for ( period (t) :
                  @sum ( grade (g) : Beta(a,r,g) * X(a,g,t) ) = @sum
( Link11 (s,a,r,t) : F1(s,a,r,t) ) ) ) );

@for ( supplier (s) :
      @for ( rawmaterial (r) :
            @for ( period (t) :
                  @sum ( Link11 (s,a,r,t) : F1(s,a,r,t) ) <=
capa1(s,r) ) ) );

!Affiliate Constraints;

@for ( affiliate (a) :
      @for ( period (t) :
            @sum ( Link4 (a,g,t) : x(a,g,t) ) + @sum ( Link4 (a,g,t)
: B(a,g,t) * Y(a,g,t) ) <= capa2(a,t) ) );

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( period (t) :
                  x(a,g,t) <= 100000 * Y(a,g,t) ) ) );

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( period (t) :
                  100*Y(a,g,t) <=  x(a,g,t) ) ) );

@for ( Link4 (a,g,t) : @BIN ( Y (a,g,t) ) );

!Warehouse Constraints;

@for ( affiliate (a) :
      @for ( grade (g) :
            @for ( period (t) :
                  X(a,g,t) = @sum ( Link12 (a,w,g,t) : F2(a,w,g,t) )
) ) );



@for ( warehouse (w) :
      @for ( grade (g) :
            @for ( period (t) | t #EQ# 1:
                  Intial(w,g,t) + @sum ( Link12 (a,w,g,t) :
F2(a,w,g,t) ) = I(w,g,t) + @sum ( Link13 (w,k,g,t) : F3(w,k,g,t) ) )
) );

@for ( warehouse (w) :
      @for ( grade (g) :
            @for ( period (t) | t #GT# 1 :
```

```
                  I(w,g,t-1) + @sum ( Link12 (a,w,g,t) : F2(a,w,g,t)
) = I(w,g,t) + @sum ( Link13 (w,k,g,t) : F3(w,k,g,t) ) ) ) );

!Customer Constraints;

@for ( grade (g) :
      @for ( customer (k) :
            @for ( period (t) :
                  @sum ( Link13 (w,k,g,t) : F3(w,k,g,t) ) = D(g,k,t)
) ) );


!Binary Constraints;


@for ( Link5 (a,h,g,t) : @BIN( Z(a,h,g,t) ) );

@for ( Link4 (a,g,t) : @BIN( alpha(a,g,t) ) );

@for ( Link4 (a,g,t) : @GIN( x(a,g,t) ) );

@for ( Link6 (w,g,t) | t #GT# 7 : I(w,g,t) = 0 );

DATA:

Bignumber = 1000000;

Cost1, Cost2, Cost3, Cost4, Profit1, Cost5, Cost6, capa1, capa2, O,
Beta, D, Intial, par, Cost7, B =
@OLE ( 'C:\Users\sari\Desktop\Sensitivity analysis\main
example\heuristics\CLP\main.xlsx' , 'Cost1' , 'Cost2' , 'Cost3' ,
'Cost4' , 'Profit1'
, 'Cost5' , 'Cost6' , 'capa1' , 'capa2' , 'O' , 'Beta' , 'D' ,
'Intial' , 'par', 'Cost7' , 'B' );

@OLE (  'C:\Users\sari\Desktop\Sensitivity analysis\main
example\heuristics\CLP\main.xlsx' , 'x' , 'I') = x , I;


ENDDATA
```

Appendix C

Sample of the Second & Third Stage Heuristics Matlab Code

```
clc
clear
close all
% Import all lists
filename = 'C:\Users\sari\Desktop\Sensitivity analysis\main
example\heuristics\matlab\main.xlsx';
% prompt=('what is the range of x?');
% range = input(prompt,'s');
range = 'c3:k27';
x = xlsread(filename,'x',range);

profit = xlsread(filename,'Profit1',range);

scost = xlsread(filename,'cost7',range);

% prompt=('what is the range of transition cost?');
% range = input(prompt,'s');
range = 'c3:k127';
Costs = xlsread(filename,'Cost5',range);

transition_matrix = xlsread(filename,'O',range);

% prompt=('what is the range of capacity?');
% range = input(prompt,'s');
range = 'b3:j7';
capacity = xlsread(filename,'capa2',range);

% prompt=('what is the range of par?');
% range = input(prompt,'s');
par = 'b3:j7';
par = xlsread(filename,'par',range);

% prompt=('what is the range of demand?');
% range = input(prompt,'s');
range = 'c3:k52';
demand = xlsread(filename,'D',range);

% prompt=('how many affiliate?');
% a = input(prompt);
a = 5;
% prompt=('how many grades?');
% g = input(prompt);
g = 5;
% prompt=('how many periods?');
% t = input(prompt);
t = 9;
% prompt=('how many customers?');
% k = input(prompt);
k=10;
affiliate_plan_matrix =
sequenceOptimization(x,Costs,transition_matrix,capacity,a,g,t);
%% Stage one Option one
count_sr11 = 0; % sequence reduction counter
comparecost = [];
flag = 0;
```

```
counta = 1;
xsetup_cost = x; % Create the initial feasible solution

ir = 1;
while ir < size(xsetup_cost,1) % loop each affiliate

    comparecost = [];
    it = t-1;
    while it > 2 % start at the last period and move backwards
        comparecost = [];
        count = 1;% used as index for the compare cost , compare
index
        ig = 1;
        while ig <= g % move through the grades
            if xsetup_cost(ig+ir-1,it)>0 % check if  the grade is
produced
                comparecost(count) = scost(ig+ir-1,it); % if so store
the cost of corresponding grade
                compareind(count,:) = [ig+ir-1,it]; % and store the
grade`s index
                count = count +1;
            end
            ig = ig+1;
        end
        % here we should skip if none

        if isempty(comparecost)
            flag = 1;
        end
        if flag
            flag = 0;
            it = it -1;
            continue % move to next period
        else
            [maxg,ind] = max(comparecost'); % retrieve the max setup
cost to choose the grade
            maxgind = compareind(ind,:); % use  the index to get the
value of the grade
            % check capacity
            sum = cumsum(xsetup_cost(ir:ir+g-1,it-1)); % get the
total production quantity %
            sum = sum(end);
            if ir == 1
                summy =
cumsum((affiliate_plan_matrix(1:g*g,it)).*transition_matrix(1:g*g,it)
);
            else
                summy = cumsum((affiliate_plan_matrix(1+g*g*(((ir-
1)/g)-1):g*g+g*g*(((ir-1)/g)-1),it)).*transition_matrix(1+g*g*(((ir-
1)/g)-1):g*g+g*g*(((ir-1)/g)-1),it));
            end
            totaltrasitionalmaterial = summy(end);
            availablecapa = capacity(counta,it-1)-sum-
totaltrasitionalmaterial;
            if availablecapa>xsetup_cost(maxgind(1),maxgind(2)) %
check if available capacity is enough
                if xsetup_cost(maxgind(1),maxgind(2)-1)>0 % check if
we are producing the same grade in the previous period
```

```
                    xsetup_cost(maxgind(1),maxgind(2)-1) =
xsetup_cost(maxgind(1),maxgind(2)-
1)+xsetup_cost(maxgind(1),maxgind(2)); % yes , move
                    xsetup_cost(maxgind(1),maxgind(2)) = 0;

                    affiliate_plan_matrix =
sequenceOptimization(xsetup_cost,Costs,transition_matrix,capacity,a,g
,t);

                    count_sr11 = count_sr11 + 1;
                    if ( it < t - 1 )
                    it = it+2;
                    end
                else

                    countb = 1; % counter to jump the affiliates
                    ir2 = 1;
                    compareprofit= [];
                    while ir2 <size(xsetup_cost,1) % check all other
affiliates for available capacity
                        sum = cumsum(xsetup_cost(ir2:ir2+g-1,it-1));
                        sum = sum(end);
                        if ir == 1
                            summy =
cumsum((affiliate_plan_matrix(1:g*g,it)).*transition_matrix(1:g*g,it)
);
                        else
                            summy =
cumsum((affiliate_plan_matrix(1+g*g*(((ir-1)/g)-1):g*g+g*g*(((ir-
1)/g)-1),it)).*transition_matrix(1+g*g*(((ir-1)/g)-1):g*g+g*g*(((ir-
1)/g)-1),it));
                        end
                        totaltrasitionalmaterial = summy(end);
                        availablecapa(countb) = capacity(countb,it-
1)-sum-totaltrasitionalmaterial;


                        indexes = maxgind(1)-(ir-1)+g*(countb-1);



                        if
availablecapa(countb)>xsetup_cost(maxgind(1),maxgind(2)) % if not
skip
                            if xsetup_cost(indexes,maxgind(2)-1)>0
% if not skip
                                compareprofit(countb) =
profit(indexes,maxgind(2)-1);
                                profitindcomp(countb,:) =
[indexes,maxgind(2)-1];
                            end
                        end

                        countb = countb+1;
                        ir2=ir2+g;
                    end


                    if isempty(compareprofit)
```

```
                        it = it -1;
                        continue

                    else
                        [maxp,ind] = max(compareprofit'); % find the
highest profit
                        maxpind = profitindcomp(ind,:);
                        xsetup_cost(maxpind(1),maxpind(2)) =
xsetup_cost(maxpind(1),maxpind(2))+xsetup_cost(maxgind(1),maxgind(2))
;
                        xsetup_cost(maxgind(1),maxgind(2)) = 0;

                        affiliate_plan_matrix =
sequenceOptimization(xsetup_cost,Costs,transition_matrix,capacity,a,g
,t);
                        count_sr11 = count_sr11 + 1;
                        if ( it < t - 1 )
                        it = it+2;
                        end
                        ir = 1;
                        counta = 1;
                    end

                end
            else

                countb = 1; % counter to jump the affiliates
                ir2 = 1;
                compareprofit= [];
                while ir2 <size(xsetup_cost,1) % check all other
affiliates for available capacity
                    sum = cumsum(xsetup_cost(ir2:ir2+g-1,it-1));
                    sum = sum(end);
                    if ir == 1
                        summy =
cumsum((affiliate_plan_matrix(1:g*g,it)).*transition_matrix(1:g*g,it)
);
                    else
                        summy =
cumsum((affiliate_plan_matrix(1+g*g*(((ir-1)/g)-1):g*g+g*g*(((ir-
1)/g)-1),it)).*transition_matrix(1+g*g*(((ir-1)/g)-1):g*g+g*g*(((ir-
1)/g)-1),it));
                    end
                    totaltrasitionalmaterial = summy(end);

availablecapa(countb) = capacity(countb,it-1)-sum-
totaltrasitionalmaterial;

                    indexes = maxgind(1)-(ir-1)+g*(countb-1);


                    if
availablecapa(countb)>xsetup_cost(maxgind(1),maxgind(2)) % if not
skip
                        if xsetup_cost(indexes,maxgind(2)-1)>0
% if not skip
                            compareprofit(countb) =
profit(indexes,maxgind(2)-1);
```

```
                            profitindcomp(countb,:) =
[indexes,maxgind(2)-1];
                        end
                    end
                    countb = countb+1;

                    ir2=ir2+g;
                end

                if isempty(compareprofit)
                    it = it -1;
                    continue
                else
                    [maxp,ind] = max(compareprofit'); % find the
highest profit
                    maxpind = profitindcomp(ind,:);
                    xsetup_cost(maxpind(1),maxpind(2)) =
xsetup_cost(maxpind(1),maxpind(2))+xsetup_cost(maxgind(1),maxgind(2))
;
                    xsetup_cost(maxgind(1),maxgind(2)) = 0;

                    affiliate_plan_matrix =
sequenceOptimization(xsetup_cost,Costs,transition_matrix,capacity,a,g
,t);
                    count_sr11 = count_sr11 + 1;
                    if ( it < t - 1 )
                    it = it+2;
                    end
                    ir = 1;
                    counta = 1;
                end

            end
        end
        it = it-1 ;
    end
    counta = counta+1;
    ir = ir + g;
end
```

Vita

Sari Hilmi Abdullah was born in 1993, in Amman, Jordan. He moved to the United Arab Emirates in 1996. He was educated in private schools and he was graduated from Al Ahliya Private School with honors.

In 2014, he graduated from University of Sharjah, his degree was a Bachelor of Science in Industrial Engineering and Engineering Management. After Graduation he joined the Engineering Systems Management Program at the American University of Sharjah where he was a Graduate Teaching Assistant.

Engineer Sari participated in the Sixth Industrial Engineering and Operation Management conference IEOM in Kuala Lumpur, Malaysia in 2016 where he presented a paper related to Capacitated Lot-Sizing with Sequence-Dependent Setup Costs.