MODELING PRODUCT RECONFIGURATION AT DISTRIBUTION CENTER LEVEL

By

JEET ARVIND TURAKHIA

Bachelor of Engineering

Shivaji University

Kolhapur, Maharashtra, India

2010

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
Master of Science
July 2013

MODELING PRODUCT RECONFIGURATION AT DISTRIBUTION CENTER LEVEL

Thesis Approved:

Dr. Manjunath Kamath

Thesis Adviser

Dr. Balabhaskar "Baski" Balasundaram

Dr. Ricki G. Ingalls

ACKNOWLEDGMENTS

" A graduate degree is not about just learning, it is about learning how to learn" are the words of my thesis adviser, Dr. Manjunath Kamath. These words of his prompted me to take up the thesis option. I want extend my gratitude to Dr. Kamath for motivating me to pursue a thesis and for guiding me throughout my MS degree. I will always be thankful for his patience in answering my questions and correcting me when I was wrong. I will always be indebted to him for his support and guidance.

I want to express my sincere thanks to Dr. Balabhaskar "Baski" Balasundaram for serving on my committee and providing his expertise in developing the mathematical formulations. I cannot begin to thank him enough for the time and energy he spent on both models, explaining them to me and making sure they were correct. I owe my improved understanding of optimization to him. My thanks are also due to Dr. Ricki G. Ingalls for his patience and constant support as a committee member.

I would like to thank Jeremy Chinn for teaching me about logistics, warehousing, and distribution centers. It is thanks to him that I was able to take my internship experience and convert it to a thesis. He was the best supervisor an intern can have. He ensured that the work environment was conducive to my progress as an employee and as a student.

I want to thank the faculty and staff at the IEM department, this thesis effort would have been difficult without their help

I am grateful to my friends Anand Govindarajan, Kumar Singarapu, Montoo Gandhi, Sunil Kumar Lakkakula, Suresh Kumar Jayaraman, Upasana Manimegali Sridhar, Vijayalakshmi Sethuraman, Samyukta Koteeswaran and Vini Singh for their

constant mental support and for always being available to lend an ear to my rants. They made my stay at Stillwater like home. I also want to thank Juan "Julie" Ma for proof reading my thesis document, helping me with LaTeXand for being a wonderful colleague. Special thanks to Sameer Ghewari for allowing me to bounce ideas off of him and for making my learning experience in the US worthwhile.

My family is the cornerstone of all success. It is their love, encouragement and support that have helped me through the toughest of times. My sister, Bhakti's unconditional love has driven me during my stay at Oklahoma State University. I do not know how to thank Tejas Deshpande; her support has let me see through many bold decisions, the thesis effort would have been difficult without her. I will forever be grateful to my parents for giving, even before I needed anything. All my work and success is dedicated to them. [1]

[1] Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: JEET ARVIND TURAKHIA

Date of Degree: July 2013

Title of Study: MODELING PRODUCT RECONFIGURATION AT

DISTRIBUTION CENTER LEVEL

Major Field: INDUSTRIAL ENGINEERING AND MANAGEMENT

Scope and method of study: This study focused on recognizing, understanding, and modeling product reconfiguration. Product reconfiguration is a process by which an in-stock product is reconfigured to meet the demand of an out-of-stock product with which it shares common parts or kits. The donor product consumes in-stock service-kit(s) for the reconfiguration process and releases kit(s) that can be used in other reconfigurations. Product reconfiguration can be employed by enterprises striving for mass customization with off-shore manufacturing facilities and local-to-demand Distribution Centers (DC). We modeled the reconfiguration process using single period Network Flow based and Integer Programming (IP) formulations and developed a prototype Decision Support System (DSS) that interfaces with the optimization models. Product reconfiguration is a relatively new field and published literature is limited. Our modeling approaches contribute to this field by providing insight into the product reconfiguration process at DC level.

Findings and Conclusions: The models aid in making reconfiguration decisions by optimally deciding which product is to be reconfigured to which other product and by consuming which kit(s). The IP formulation is an aggregate planning model and the network flow based formulation specifies which kit inventory to consume. Post processing algorithms were necessary to construct reconfiguration decisions from the solution to the network flow model. The DSS acts as an interface between the model data and optimization software; final reconfiguration decision from the network model is also displayed in the DSS. This research has established the need for and the applicability of optimization models in the product reconfiguration context; it has also highlighted the importance of a DSS in supporting reconfiguration operations and in developing and refining the underlying decision-making models.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A distribution center (DC) is typically the final point in a supply chain from where the product is delivered to a retailer or a customer. With manufacturing activities moving offshore, the resulting increase in lead time has placed a lot of importance on inventory management within the warehouse and DC. Storage at the demand site acts as a buffer for a manufacturing enterprise against demand uncertainty and long lead times. Literature supports the fact that costs incurred in operating and maintaining warehouses and DCs are substantial and several papers have been published on the analysis of warehouse and distribution channels.

A DC can be considered to be a warehouse, but with a few additional services such as assembly, sorting, and packaging. A warehouse facilitates static storage and for the purpose of this document, assembly and other value added operations are necessary, which is typical of a DC, hence all operations are considered to transpire in the DC.

As enterprises become more customer centric, they try to increase the product variety and strive for mass customization [Venkatesh and Swaminathan, 2004]. An enterprise would like to hold the least amount of on-hand inventory and with it meet the most demand; in order to accomplish this forecasting demand is essential. Forecasted demand almost always deviates from the actual demand. The actual demand becomes clearer as we near the final point of sale and in this study we assume it to be the DC.

The DC, in case of this thesis, is responsible for sale of end products and sub-

assemblies to the customer. Examples of such sub-assemblies are service kits for the products and will be referred to as kits or service-kits for remainder of this document. The DC is also equipped to perform partial production operations like dis-assembly and assembly of end-products.

Decreasing costs in developing economies is cited as a major factor in the selection of a manufacturing facility. Setting up and operating a manufacturing facility in developed economies can be a costly affair. The major drawback of setting up facilities offshore in developing economies is the increase in lead times because of increased transportation time. This increased lead time causes aggravated demand uncertainty. Production planning at the manufacturing facility requires demand forecasts well ahead of time to exploit the production capacity for mass production. As production is underway and products are being shipped, the possibility of change in the customer order increases and this change can be in regard to the product options or product quantity. A reconfiguration strategy at the DC will help in mitigating the uncertainty in customer demand while not affecting the production planning at the manufacturing facility.

1.1 DC with Product Reconfiguration

Reconfiguration can be utilized when there is demand for a product that is out of stock, but another product from the same product family has excess inventory. If the actual demand deviates from its forecast, products can be reconfigured to meet this new demand. In case demand exists for an out-of-stock product, instead of forfeiting demand, a product from its product family can be reconfigured to be sold as the out-of-stock product. Both end products and kits can be directly sold to the customer and constitute a direct sale.

The DC stocks end products and service kits. End products can be said to be

grouped into families; to qualify to be part of a family, two end products must share at least one common kit. For example, consider two products, product A and product B, each of which contains three kits.

- Product A is made up of kit 1, kit 2 and kit 3 and;
- Product B is made up of kit 1, kit 2 and kit 4.

Since both the products share kit 1 and kit 2 they are assumed to be part of the same product family. Product B can be converted to product A by replacing kit 4 by kit 3 by what is called in this document as *reconfiguration*.

The item that donates itself for conversion is called the *donor item*. It also is responsible for a *reclaimed-kit* being put back into the inventory. In the above example, product B is the donor item and kit 4 is generated as a reclaimed-kit. A reclaimed-kit cannot satisfy the demand for a new service-kit.

1.2 Related Supply Chain Strategies

A supply chain strategy that is similar to the one considered in this study is the PUSH-PULL strategy [Simchi-Levi et al., 2004]. In the initial stages, the product is pushed through the supply chain based on a forecast. If customer demand is aligned with the PUSH forecast the product is shipped as-is; otherwise, the product is reconfigured to realize the variation in customer demand thus following the PULL strategy. The strategy of reconfiguring the product only at the time of clear visibility of the demand can be viewed as the postponement strategy or delayed differentiation strategy. Postponement refers to delaying the point in supply chain where the final personality of the product is finalized [Swaminathan and Lee, 2003]; preferably till customer demand is realized. Our research effort deals with stocking ready-to-ship end products at the DC and modifying the product, if necessary, to meet the customer

demand. The structural modification of the product results in one or more kits being put back into the inventory and at this point the research effort deviates from an ideal postponement strategy.

The reconfiguration strategy can also be seen as a hybrid between the make-to-stock and make-to-order manufacturing environments. In the first part of the supply chain the make-to-stock strategy is followed at the manufacturing facility and product is shipped to the DC. If customer demand is aligned with the make-to-stock forecast, the product is shipped as is. If the customer demand deviates from the forecast, the product is reconfigured to align with the demand thus following the make-to-order philosophy [Simchi-Levi et al., 2004].

Postponement or delayed product differentiation, make-to-stock, make-to order, push-pull strategies aim at fulfilling all the possible customer demand while limiting the inventory holding cost. The supply chain strategy employed for the means of this research effort could be considered a hybrid strategy containing elements of all these strategies.

The strategy of reconfiguration deviates from remanufacturing strategy in a closed loop supply chain in the following manner: In contrast to reclaiming products by dismantling and separating the "good" quality ones, this strategy will remove kits of brand new end-products and replace them to fulfill a different order. Pagh and Cooper [1998] stated that risk or uncertainty can be reduced or fully eliminated by postponing manufacturing or logistics operations till the customer completely commits to the order.

1.3 Outline of the Document

The rest of this thesis document is organized as follows. Chapter 2 reviews relevant literature that has been published and explains how it relates to this thesis work. Chapter 3 presents the problem statement, research objectives, and the re-

search contribution. Chapter 4 presents the network flow-based and integer programming approaches developed to model product reconfiguration. Chapter 5 explains the development of a decision support system that incorporates the two optimization models. Chapter 6 explains the test data used, its generation and explains how both models performed. Chapter 6 also presents the results and insights obtained from testing. Chapter 7 concludes the document with a summary of the research conducted and then discusses future avenues for research.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a brief review of the literature related the postponement strategy, closed loop supply chains and mass customization. Section 2.1 explains the manufacturing postponement strategy and its relation to the problem being studied. Section 2.2 draws parallels between closed loop supply chains and product reconfiguration and section 2.3 discusses the importance of design for product families in accordance with mass customization in the product reconfiguration scenario. Section 2.4 describes the literature that focusses on the product in the selection of a supply chain strategy. Finally, the chapter describes how research approaches employed in these published studies relate to this research effort.

2.1 Manufacturing Postponement Strategy

According to Pagh and Cooper [1998], under the manufacturing postponement strategy the final manufacturing operations are performed at some point downstream in the supply chain. Davis [1993] analyzes the Hewlett-Packard Desk-Jet printer supply chain. The strategy involves deferring configuring the product till the final customer order is received. This configuration takes place at local distribution centers. Davis [1993] notes that incorporating this strategy causes a slight increase in manufacturing cost but the safety stock is reduced. The supply chain strategy employed for this thesis requires that the local DC stock ready-to-ship end products; but these products can be reconfigured in order to realize a customer demand that may have changed. By stocking a configurable product and reconfiguring only when a customer order is

received, the supply chain strategy involved in this thesis is in line with this thinking. The reader is also referred to Van Hoek [2001] for an in depth literature review of the postponement strategy.

2.2 Remanufacturing and Closed Loop Supply Chain

Closed loop supply chains are defined as "Supply chains that are designed to consider the acquisition and return flows of products, reuse activities, and the distribution of the recovered products" [Lambert, 2008]. Research that is somewhat relevant to the reconfiguration problem has been published under the remanufacturing category.

The framework by Van der Laan et al. [1999] is similar to the problem being studied in this thesis. The paper focusses on the effect of lead time and its variability on the remanufacturing system defined by Van der Laan and Salomon [1997]. Once modules are deemed to be of "good quality" they are used as product components on new products and sold to the end customer. In the case of this thesis, the service-kits of end products can be used for reconfiguration. Ready-to-ship end products are reconfigured using service kits, the service kits are comparable to the "good quality" modules and the ready-to-ship end products are new products as defined by Van der Laan et al. [1999]. In principle, both the paper [Van der Laan and Salomon, 1997] and the thesis share a common structure in the manufacturing part.

2.3 Mass Customization

From a product design point of view, this research approach supports designing product families instead of individual products. The perspective of designing for product families is common with the mass customization approach [Sabin and Weigel, 1998]. Although the product design part is not a focus of this research effort, its existence in the design phase is essential for smooth functioning of the reconfiguration methodology.

For a product family, there can be various ways in which one product can be reconfigured to be sold as another one. This research focusses on the operational aspect where a manager needs to determine which reconfiguration method results in what cost. Method refers to which product to reconfigure to meet which demand.

2.4 Supply Chain Strategy with Respect to the Product

Substantial research has been published in choosing the correct strategy for a supply chain with the focus on the product [Aitken et al., 2003, Lee, 2002, Pagh and Cooper, 1998, Venkatesh and Swaminathan, 2004]. Aitken et al. [2003] focus on the need to align the supply chain strategy of an enterprise with the stage of product life cycle. They studied the re-engineering of supply chain as performed by a UK lighting company to accommodate the impact of product life cycle on the supply chain.

Lee [2002] classifies choosing the supply chain for a product based on supply and demand characteristics. Lee [2002] describes how products are classified based on the uncertainty framework and then goes on to list what strategy has been historically used for each type of product.

Venkatesh and Swaminathan [2004] describe how postponement can be an applied to managing product variety. They talk about how companies like Motorola, Hewlett-Packard, and Xilinx have utilized postponement. They also explain the product and process enablers that help in developing an effective postponement strategy.

The direct effect of postponement and product reconfiguration on operations has not been addressed by the literature. This research has developed models that will help in the operational practices of a postponement supply chain strategy with product reconfiguration.

This chapter has reviewed the literature from related fields and provided insight into how each one is related to the research topic of this thesis. There is no model or framework in the current literature that can directly or by modification be employed to

address the problem of product reconfiguration. It is evident that additional research is necessary in the area of product reconfiguration which will allow for better decision making from both tactical and operational perspectives.

CHAPTER 3

RESEARCH STATEMENT

This chapter describes the research statement of this thesis effort. Section 3.1 describes the motivating factor and an industry example that led to the topic of this research. Section 3.2 includes the problem statement, section 3.3 describes the research objectives, and section 3.4 defines the scope that limits this research effort. Section 3.5 describes deliverables of the research effort and the contribution that it will make to the field of Supply Chain Management.

3.1 Motivating Example

The topic studied in this thesis was motivated by the author's summer internship experience in an enterprise that has a DC operating and servicing the Americas. As a global manufacturing enterprise with manufacturing facilities located offshore, long lead times were experienced for delivery of all their end products and service-kits to the American DC. The strategy of product reconfiguration to meet a deviation in demand was practised as the primary supply chain strategy. While working for the enterprise the author observed the reconfiguration strategy employed by the enterprise.

The enterprise stocked the DC based on a forecast and products were reconfigured if the realized demand deviated from the forecast. Every time reconfiguration was necessary, the decision of how to meet demand was placed on the warehouse manager and the customer service representative (CSR). In this process, the systemic inventory of products and kits was verified and cross-referenced with demand and

the reconfiguration method was decided. The reconfiguration method refers to which donor item and which service kit(s) are consumed to meet the demand. Kits released from reconfiguration were placed into inventory; and their systemic transaction ensured correct augmentation of inventory. These were then consumed only for other reconfiguration requests.

As a part of this process, a document with a list of items to be picked for reconfiguration, the type of reconfiguration to be performed and the locator based putaway of items was generated for the technicians. As a single reconfiguration method might involve multiple units of a type of donor item and service kit(s), the possibility of a mismatched putaway increases. But since the putaway is specified by the aforementioned document, a mismatch between the physical inventory and systemic inventory can also be created, called inventory record inaccuracy as coined by Rinehart [1960] and studied by DeHoratius and Raman [2008]. This issue has not been addressed in our research effort.

Currently if products need to reconfigured, the manager and CSR take a reconfiguration decision based on their knowledge of products and product families. This research effort believes that the decision of how to meet a demand and the product reconfiguration decision, if any, can be made easier and smoother by means a decision support system (DSS) driven by an optimization model. By developing the model and DSS, the task of deciding how to meet a customer demand will be given to the model / DSS and not the manager.

3.2 Problem Statement

Little research exists that directly addresses the problem of cost-based product reconfiguration. No model(s) or directly applicable analysis technique has presented itself during the time of writing this document that aims to optimally solve a problem of this nature.

The problem statement is as follows.

To support the reconfiguration decision making process by means of a DSS that is based on optimization model(s).

3.3 Research Objectives

The specific objectives of the research are as follows:

1. To develop optimization models in order to facilitate better decision making in the context of product reconfiguration.

    - A network flow-based model and an integer programming model were developed to represent the product reconfiguration methodology to meet customer demand and to determine the reconfiguration decision and its cost.

2. To develop a prototype DSS environment to support product reconfiguration decisions.

    - A Microsoft® Excel / VBA based decision making environment was built by incorporating the optimization model to analyze the cost of reconfiguration decisions and the ending inventory of end products and kits.

3.4 Scope and Limitations

The scope of the research was limited by the following assumptions:

1. The optimization models depict a single period instance of the problem.

2. The models were tested using only randomly generated input data.

3. Processing times at the DC, deviation and evolution of forecast were not considered.

4. The inbound, shelving, order picking, and outbound processes of the DC were assumed to be running in accordance with the incoming orders and were not the focus of the research.

3.5 Deliverables and Contribution

This section describes the deliverables from this research effort and expected contribution to the industrial engineering body of knowledge.

3.5.1 Deliverables

The two primary deliverables for this research are given below.

1. Network flow-based and integer programming models representing the consumption of products and kits to meet demand.

2. A Microsoft® Excel / VBA based prototype Decision Support System (DSS) to support reconfiguration decisions

3.5.2 Research Contribution

A contribution of this research is in the development of a generic representation framework of product reconfiguration in a unit-load based discrete manufacturing environment such as internal combustion engines, AC generators and electronic goods with interchangeable components. The DSS resulting from this research effort will aid in tactical and operational planning. From a tactical perspective, compatible and or obsolete inventory can be reconfigured and sold as per customer specifications. From an operational perspective, if reconfiguration is possible, then reconfiguration decisions can be delegated to the shop floor for order fulfillment.

This research contributes to the field of supply chain management by developing optimization models that can be applied to a class of reconfiguration operations

at DCs. One of the model exploits the minimum cost network flow-based model framework to capture reconfiguration decisions.

This chapter discussed the motivating factor behind this research effort. It also stated the problem and research objectives. The outcome of the research effort and its limitations have also been described and specified. Finally, the chapter also described the contribution of this research to field of supply chain management.

CHAPTER 4

MODELING APPROACH

This chapter starts with an example to illustrate the modeling approach employed in this research effort. It is a simplified example and does not include all the parameters and decision variables that are part of the actual models; but has enough details for the reader to understand the workings of the optimization models. Section 4.2 describes the optimization models utilized to model product reconfiguration, it then goes on to explain in detail the notation, formulation, and limitation of each model.

4.1 Three-Product Example

Suppose the product line consists of three products - product 1, product 2, and product 3

- Product 1 is made up of kit 1, kit 3 and kit 4
- Product 2 is made up of kit 2 and kit 5 and;
- Product 3 is made up of kit 1, kit 2, kit 3 and kit 5.

Since products 1 and 3 share kit 1 and kit 3, they are part of a product family and hence compatible. Products 2 and 3 share kit 2 and kit 5, they form another product family and are compatible with each other. In this example, products 1 and 2 are in two distinct product families.

4.1.1 Product Reconfigurations

1. Product 1 can be reconfigured to be sold as product 3.

- Product 1 is the donor product.
- Reconfiguring product 1 to product 3 results in kits 2 and 5 being consumed from reclaimed kit or new kit inventory; kit 4 being generated as a reclaimed kit.

2. Product 2can be reconfigured to be sold as product 3.
    - Product 2 is the donor product.
    - Reconfiguring product 2 to product 3 results in kit 1 and kit 3 being consumed from reclaimed kit or new kit inventory. No reclaimed kits are generated from this reconfiguration.
3. Product 3 can be reconfigured to be sold as product 1 and product 2.
    - Product 3 is the donor product.
    - Reconfiguring product 3 to product 1 results in kit 4 being consumed from reclaimed or new kit inventory; kit 2 and kit 5 being generated as reclaimed kits.
    - Reconfiguring product 3 to product 2 results in no new reclaimed kit being consumed; but kit 1 and kit 3 are generated as reclaimed kits. Cases such as this where no kits are consumed are not considered in this study.

4.2 Optimization Models

The process of reconfiguration was modeled using the following two methods.

- Network flow-based formulation and;
- Integer programming based formulation

As mentioned before we consider decisions made in a single period only. Ideally, the model will be executed at the start of a work day to determine how the demand for that day should be met. A fully notated three-product example can be found in Appendix A.

4.2.1 Network Flow-based Formulation

A large number of applications can be modeled using network flow-based models [Ahuja et al., 1993, Jensen and Bard, 2003]. The network flow-based problem has a linear objective and can be efficiently solved by the simplex method [Ahuja et al., 1993]. We use a modified version of the minimum cost network flow-based problem formulation to model the reconfiguration process. The objective is to minimize the cost of reconfiguration incurred to meet customer demand. The inventory of products and kits are the supply/origin nodes, the demand of the products and kits are demand/destination nodes, and inventory of reclaimed kits is represented as capacitated transshipment nodes. Since the inventory of new kits can be used for both direct sale and reconfiguration requests, two copies of the node set were created. The on-hand inventory of new kits acts as the capacity on these nodes. Every node in the first node set shares its capacity with its corresponding node in the second node set. Set one is used exclusively for meeting direct sale requests for new kits and set two is used for reconfiguration requests. This shared capacity can be thought of as the capacity of a single node in the following manner.

The outflow from a node in set one and from its copy in set two could be directed to a new node, whose capacity is the shared capacity of the two nodes. This way the nodes in set one and set two would no longer be capacitated and the new node would act as a capacitated transshipment node. Our formulation does not explicitly consider the generation of reclaimed kits from a reconfiguration process; but its implementation

computes the number of reclaimed kits generated via post processing(see section 5.2.1).

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ represent the set of m products and $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$ represent the set of n kits.

The bill of materials defined by $\mathcal{B}(p) \subseteq \mathcal{K}$ is the set of kits used by $p \in \mathcal{P}$

Let $G = (N, A)$ be a directed graph with a set of N nodes and set of A directed arcs.

1. **Definition of nodes**

$$N = N_p^+ \cup N_k^+ \cup N_p^- \cup N_k^- \cup N_k^R \cup N_k^C \cup N_k^G \cup N^D$$

Where,

- $N_p^+ = \{p_1^{(1)}, p_2^{(1)}, \ldots, p_m^{(1)}\}$; Set of product supply nodes
- $N_k^+ = \{k_1^{(1)}, k_2^{(1)}, \ldots, k_n^{(1)}\}$; Set of kit supply nodes
- $N_k^- = \{k_1^{(2)}, k_2^{(2)}, \ldots, k_n^{(2)}\}$; Set of kit demand nodes
- $N_p^- = \{p_1^{(2)}, p_2^{(2)}, \ldots, p_m^{(2)}\}$ Set of product demand nodes
- $N_k^R = \{k_1^{(3)}, k_2^{(3)}, \ldots, k_n^{(3)}\}$; Set of capacitated reclaimed kit nodes:
- $N_k^C = \{k_1^{(4)}, k_2^{(4)}, \ldots, k_n^{(4)}\}$; Set **one** of shared capacitated kit supply node copy

- $N_k^G = \{k_1^{(5)}, k_2^{(5)}, \ldots, k_n^{(5)}\}$; Set **two** of shared capacitated kit supply node copy:

- N^D represents a dummy node

2. **Definition of Arcs**

   (a) Direct sale of a kit:

   $(k_i^{(1)}, k_i^{(4)}), (k_i^{(4)}, k_i^{(2)}) \in A$

   $\forall i = 1, 2, \ldots, n$

   Illustration:

   $N_k^+ \rightarrow N_k^C \rightarrow N_k^-$

   (b) Direct sale of a product:

   $(p_i^{(1)}, p_i^{(2)}) \in A$

   $\forall i = 1, 2, \ldots, m$

   Illustration:

   $N_p^+ \rightarrow N_p^-$

   (c) Sale of a product by reconfiguration:

   Product i is reconfigured and sold as product j by consuming kits $\{k_{v_1}, \ldots, k_{v_r}\}$

   Consider $p_j^{(2)} \in N_p^-$ and $p_i^{(1)} \in N_p^+$

   Such that:

   - $i \neq j$
   - $\mathcal{B}(p_j^{(2)}) \cap \mathcal{B}(p_i^{(1)}) \neq \emptyset$ and;
   - $\mathcal{B}(p_j^{(2)}) \setminus \mathcal{B}(p_i^{(1)}) = \{k_{v_1}, \ldots, k_{v_r}\}$

     $r =$ Total number of kits required for reconfiguration

Then,

For $r = 1$

- $\left(p_i^{(1)}, k_{v_1}^{(3)}\right) \text{and} \left(k_{v_1}^{(3)}, p_j^{(2)}\right) \in A$
  Illustration:
  $N_p^+ \rightarrow N_k^R \rightarrow N_p^-$

- $\left(p_i^{(1)}, k_{v_1}^{(5)}\right) \text{and} \left(k_{v_1}^{(5)}, p_j^{(2)}\right) \in A$
  Illustration:
  $N_p^+ \rightarrow N_k^G \rightarrow N_p^-$

For $r \geqslant 2$;

- $\left(p_i^{(1)}, k_{v_1}^{(3)}\right) \text{and} \left(p_i^{(1)}, k_{v_1}^{(5)}\right)$
  $\forall l = 1, 2, \ldots (r-1)$
  $\left(k_{v_l}^{(3)}, k_{v_{(l+1)}}^{(3)}\right), \left(k_{v_l}^{(5)}, k_{v_{(l+1)}}^{(5)}\right), \left(k_{v_l}^{(3)}, k_{v_{(l+1)}}^{(5)}\right) \text{and} \left(k_{v_l}^{(5)}, k_{v_{(l+1)}}^{(3)}\right) \in A$
  $\left(k_{v_r}^{(5)}, p_j^{(2)}\right) \in A \text{and} \left(k_{v_r}^{(3)}, p_j^{(2)}\right) \in A$

  Illustration:
  $N_p^+ \rightarrow N_k^R \rightarrow N_p^-$
  $N_k^G \rightarrow N_k^G$
  $N_k^R \rightarrow N_k^R$
  $N_k^G \rightarrow N_k^R$
  $N_k^R \rightarrow N_k^G$
  $N_p^+ \rightarrow N_k^G \rightarrow N_p^-$

(d) In order to allow the demand to exceed supply an arc exists between the dummy node and all product and kit demand nodes.

- $(i, j) \in A$

$\forall i \in N^D, j \in N_p^-$ and $j \in N_k^-$

3. **Supply capacity and demand requirement**

   - Each node $i \in \{N_p^+, N_p^-, N_k^+, N_k^-\}$ is associated with a number $B(i)$ indicating its supply or demand depending on $B(i) > 0$ or $B(i) < 0$
   - Node $i \in \{N_k^C\}$ and its equivalent(copy) node $i \in \{N_k^G\}$ are associated with a number $T(i)$, indicating their shared (node) capacity. This number corresponds to the inventory available for that new kit
   - Each node $i \in \{N_k^R\}$ is associated with a number $U(i)$, indicating its (node) capacity. This number corresponds to the inventory available for that reclaimed kit.

4. **Costs**

   The following types of costs are included in the model. Refer table 4.1 for arc costs.

   - **Cost of reconfiguration, γ:** Cost of consuming a single unit of a reclaimed kit for a reconfiguration.
   - **Penalty cost, M^*:** M^* is some sufficiently large number. M^* is one more than the largest reconfiguration cost. The largest reconfiguration cost is, $\forall i \in \mathcal{P}, j \in \mathcal{P}$ such that $\mathcal{B}(j) \cap \mathcal{B}(i) \neq \emptyset$ and $i \neq j$,

     $$\text{Largest reconfiguration cost} = \theta * \gamma * \text{Max}\big(|\mathcal{B}(j) \setminus \mathcal{B}(i)| \big)$$

     where,

     θ = constant, $\theta > 1$;

     β = constant $\theta > 1$. (Table 4.1)

Origin i	Destination j	Cost
N_p^+	N_p^-	$\beta * \gamma$
N_k^+	N_k^C	$\beta * \gamma$
N_k^C	N_k^-	$\beta * \gamma$
N_p^+	N_k^R	γ
N_k^R	N_k^R	γ
N_k^R	N_p^-	γ
N_p^+	N_k^G	$\theta * \gamma$
N_k^G	N_k^G	$\theta * \gamma$
N_k^G	N_p^-	$\theta * \gamma$
N_k^G	N_k^R	γ
N_k^R	N_k^G	$\theta * \gamma$
N^D	N_p^-	$M^* + 1$
N^D	N_k^-	M^*

Table 4.1: Cost per Unit of Flow on an Arc

5. The optimization problem can be stated as follows.

   Let x_{ij} and α_{ij} represent the flow and cost respectively between any node $i \in N$ and $j \in N$

$$Minimize \sum_{(i,j)\in A} x_{ij} * \alpha_{ij}$$

   Subject to:

$$\sum_{i:(j,i)\in A} x_{ji} - \sum_{i:(i,j)\in A} x_{ij} \leq B(j) \qquad \begin{matrix}\forall j \in N_k^+, \\ \forall j \in N_p^+\end{matrix} \tag{4.1}$$

$$\sum_{i:(i,j)\in A} x_{ij} - \sum_{i:(j,i)\in A} x_{ji} \geq -B(j) \qquad \begin{matrix}\forall j \in N_k^-, \\ \forall j \in N_p^-\end{matrix} \tag{4.2}$$

$$\sum_{i:(j,i)\in A} x_{ji} \leq U(j) \qquad \forall j \in N_k^R \tag{4.3}$$

$$\sum_{i:(j,i)\in A} x_{ji} = \sum_{i:(i,j)\in A} x_{ij} \qquad \begin{matrix}\forall j \in N_k^R, \\ \forall j \in N_k^C \\ \forall j \in N_k^G\end{matrix} \tag{4.4}$$

$$\sum_{i:(j,i)\in A|j=k_l^{(3)}} x_{ji} + \sum_{v:(u,v)\in A|u=k_l^{(5)}} x_{uv} \leq T(j) \qquad \forall l = 1, 2, \ldots, n \tag{4.5}$$

$$\sum_{i:(j,i)\in A|j=p_s^{(1)}} x_{ji} = \sum_{i:(i,j)\in A|j=p_s^{(2)}} x_{ij} \qquad \forall s = 1, 2, \ldots, n \tag{4.6}$$

The formulation ensures that,

- Constraint 4.1: For all kit and product supply nodes the sum of outflow less the sum of inflow is utmost the supply quantity available at the node, i.e. the flow cannot exceed the supply available.

- Constraint 4.2: For all kit and product demand nodes the sum of inflow less the sum of outflow is at least equal to the demand at the node, i.e. meet as much demand as possible

- Constraint 4.3: Each node representing the reclaimed kit supply is capacitated. This constraint ensures that the sum of outflow from this node will only be as large as its capacity.

- Constraint 4.4: The sum of inflow into each of the reclaimed kits nodes, copy of new kit supply nodes, and second copy of new kit supply nodes should equal the outflow from that node so as to ensure the node does not generate a flow or consume flow.

- Constraint 4.5: The supply of new kits is shared for direct sale and for reconfiguration. The sum of outflow from any node in N_k^C and its corresponding copy in N_k^G is at least equal to the capacity (inventory / availability) of the new kits.

- Constraint 4.6: The final constraint ensures that only products are reconfigured to be sold as other products, meaning, kits should not be sold as products. This is ensured by forcing the sum of outflow from supply nodes of products to equal the sum of inflow into demand nodes of products.

Limitations of the Network Flow-based Formulation

Solution to the network flow-based model may contain reconfigurations that are not valid to the original problem. The original problem refers to the real world that is under consideration. Arcs existing between two nodes can create a path allowing a product to meet the demand of a product with which it is not compatible; this is an *invalid reconfiguration*. The model may also allow a product to consume less than necessary number of kits and meet demand of a compatible product; this is an *invalid path / flow* from the perspective of the original problem.

1. Invalid Reconfiguration

   The model allows for any product $i \in N_p^+$ to be reconfigured to any product $v \in N_p^-$ even if $\mathcal{B}(i) \cap \mathcal{B}(v) = \emptyset$ under the following conditions:

- A path exists for reconfiguring product $i \in N_p^+$ to product $j \in N_p^-$ as $\mathcal{B}(i) \cap \mathcal{B}(j) \neq \emptyset$ via node $k \in \mathcal{B}(j) \setminus \mathcal{B}(i)$
- A path exists for reconfiguring product $u \in N_p^+$ to product $v \in N_p^-$ as $\mathcal{B}(u) \cap \mathcal{B}(v) \neq \emptyset$ via node $k \in \mathcal{B}(v) \setminus \mathcal{B}(u)$
- $\mathcal{B}(j) \cap \mathcal{B}(v) \neq \emptyset$ and $k \in \mathcal{B}(j) \cap \mathcal{B}(v)$

2. Invalid Path

   The model allows for any product $i \in N_p^+$ to be reconfigured to a product $j \in N_p^-$ with which it is compatible by using fewer kits than necessary to ensure a complete reconfiguration. Consider the reconfiguration of product i to j, where $\mathcal{B}(i) = \{k_3, k_4\}$ and $\mathcal{B}(j) = \{k_1, k_2, k_4\}$. Also consider the reconfiguration of product u to product j, where $\mathcal{B}(u) = \{k_2, k_4\}$

   - Product i should consume $\{k_1, k_2\} \in \mathcal{B}(j) \setminus \mathcal{B}(i)$ to be reconfigured to j
   - Product u should consume $\{k_1\} \in \mathcal{B}(j) \setminus \mathcal{B}(u)$ to be reconfigured to j

   These conditions create two paths from product i to product j. The first path represents the consumption of k_1 and k_2. The second path represents consumption of k_1, this path is created due to the compatibility of product u and product j. The second path is "cheaper" and hence a flow along this path will allow product i to consume only kit k_1 and be reconfigured to product j, which in the original problem will not happen.

Because of these limitations feasibility of the model solution to the original problem cannot be guaranteed. This issue has been partially addressed by post processing the final solution; the post processor will only trace a valid path between two compatible products. Another post processor prints the number of different kits that are necessary for the reconfiguration next to the solution per the first post processor. The post processing algorithms, their necessity, and logic are described in section 5.2.1.

These post processors coupled with human intervention can potentially result in a feasible solution to the original problem, but the optimality of this solution cannot be guaranteed. Furthermore, the solution from the post processor and human intervention may generate a list of kits required for reconfigurations whose inventory may be insufficient to begin with. This means that the solution after post processing and human intervention may still be infeasible for the original problem. This was a major motivating factor to develop the integer programming formulation explained in the following section.

4.2.2 Integer Programming Formulation

An integer programming (IP) formulation was developed because of the limitations of the the network flow-based model described in the previous section. The IP solution ensures that a product is converted only to one it is compatible with and by consuming the correct number of kits. However, the current IP formulation is an aggregate planning model in terms of the consumption of the kits. It does not differentiate between the type of kits (new or reclaimed) being consumed by a reconfiguration. This is explained in more detail later in this section.

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ represent the set of m products, $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$ represent the set of n kits, and $\mathcal{K}' = \{k'_1, k'_2, \ldots, k'_n\}$ represent the set of n reclaimed kits.

The bill of materials (BOM) is defined by $\mathcal{B}_p \subseteq \mathcal{K}$; it is the set of kits used by $p \in \mathcal{P}$

Let S_i and D_i denote the supply quantity for all $i \in \mathcal{P}$ and $i \in \mathcal{K}$ and S'_i the supply quantity of reclaimed kit $i \in \mathcal{K}'$.

The decision variables are as follows.

X_{ij} = Number of units of $i \in \mathcal{P}$ being reconfigured into $j \in \mathcal{P}$; for all $i, j \in \mathcal{P}$ and $\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset$

Z_k = Units of kit k used in reconfiguration. $k \in \mathcal{K}$.

W_k = Units of unfulfilled demand of kit $k \in \mathcal{K}$.

U_p = Units unfulfilled demand of product $p \in \mathcal{P}$.

The IP formulation is given below.

$$Minimize \sum_{i,j \in \mathcal{P} | \mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset} C_{ij} * X_{ij} + \sum_{i \in \mathcal{P}} \alpha_i * U_i + \sum_{k \in \mathcal{K}} \beta_k * W_k$$

Subject to:

$$\sum_{i \in \mathcal{P} | \mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset} X_{ij} + U_j = D_j \qquad \forall j \in \mathcal{P} \qquad (4.7)$$

$$\sum_{j \in \mathcal{P} | \mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset} X_{ij} \leq S_i \qquad \forall i \in \mathcal{P} \qquad (4.8)$$

$$\sum_{\substack{i,j \in \mathcal{P} | (\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset, \\ k \in \mathcal{B}_j \setminus \mathcal{B}_i, i \neq j)}} X_{ij} \leq S_k + S'_k + \sum_{\substack{i,j \in \mathcal{P} | (\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset, \\ k \in \mathcal{B}_i \setminus \mathcal{B}_j, i \neq j)}} X_{ij} \qquad \forall k \in \mathcal{K} \qquad (4.9)$$

$$\sum_{\substack{i,j \in \mathcal{P} | (\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset, \\ k \in \mathcal{B}_j \setminus \mathcal{B}_i, i \neq j)}} X_{ij} - S'_k - \sum_{\substack{i,j \in \mathcal{P} | (\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset, \\ k \in \mathcal{B}_i \setminus \mathcal{B}_j, i \neq j)}} X_{ij} \leq Z_k \qquad \forall k \in \mathcal{K} \qquad (4.10)$$

$$W_k \geq D_k - S_k + Z_k \qquad \forall k \in \mathcal{K} \qquad (4.11)$$

$$Z_k \geq 0 \qquad \forall k \in \mathcal{K} \qquad (4.12)$$

$$W_k \geq 0 \qquad \forall k \in \mathcal{K} \qquad (4.13)$$

$$X_{ij} \geq 0 \qquad \forall i, j \in \mathcal{P} \qquad (4.14)$$

The objective function is to minimize the sum of the following three costs:

1. Cost of reconfiguration
2. Cost of lost sale of products
3. Cost of lost sale of kits

The formulation ensures that:

1. Constraint 4.7: For all products j, if i is compatible with j, the sum of X_{ij} is the demand of product j that is satisfied. U_j is the unfulfilled demand of product $j \in \mathcal{P}$ and loss of sale is penalized by α_j in the objective function.

2. Constraint 4.8: For all products i, if i is compatible with j, the sum of X_{ij} is the number of units of product i consumed to satisfy any compatible product's demand and this cannot exceed the supply of product i.

3. Constraint 4.9: For all kits k, if k is consumed by reconfiguring a compatible product i to j the sum of X_{ij} gives the number of kit k consumed; and this consumption cannot exceed the total supply of kit k from new kit supply, reclaimed kit supply, and reclaimed kits generated from other reconfigurations.

4. Constraint 4.10: For all kits k, Z_k denotes the kits consumed in reconfiguration. This constraint represents a linearization constraint.

5. Constraint 4.11 : The fulfillment of kit demand is modeled by penalizing each unfulfilled demand of kit $k \in \mathcal{K}$ by β_k. The difference between S_k and Z_k represents the kits leftover after reconfiguration that can be used to meet that demand D_k. The sum of D_k and Z_k less S_k represents the kits left over after meeting all possible demand requests and is denoted by W_k. To ensure the demand does not exceed supply W_k is constrained to be greater than or equal to the sum of D_k and Z_k less S_k.

Costs

1. Cost of reconfiguration is product of θ(a constant), γ, which is the cost of consuming a kit and the number of kits consumed plus one. Hence, $\forall i \in \mathcal{P}, j \in \mathcal{P}$ such that $\mathcal{B}(j) \cap \mathcal{B}(i) \neq \emptyset$

$$C_{ij} = \theta * \gamma * \big(|\mathcal{B}_j \setminus \mathcal{B}_i| + 1 \big)$$

This cost is equivalent to consuming a new kit in the network flow-based model.

2. The direct sale of kits is modeled by adding a penalty for not fulfilling a kit demand. The smallest penalty for not fulfilling the direct sale of a product has to be greater than the largest penalty for not meeting a kit demand. If not, the model will choose to meet kit demand over product demand. When the penalty for not meeting a product demand is larger than that of a kit, the model will force meeting product demand by means of reconfiguration instead of meeting a kit demand. This condition places a priority on meeting product demand over kit demand.

3. Similar to the network flow-based formulation, a reconfiguration involving no consumption of kits is not permitted. i.e. when $\mathcal{B}_j \subseteq \mathcal{B}_i$, $X_{ij} = 0$

Limitations of the IP based Formulation

1. The integer programming model is an aggregate planning model. It does not specify which type of kit (new or reclaimed) has been consumed for a reconfiguration. Unlike the network flow-based model which specifies a path that translates to consumption of kits, this data is not directly available from the solution to the IP formulation.

2. Cost of consuming a new kit equals cost of consuming a reclaimed kit. Hence, a direct comparison of the two models based solely on the objective function values cannot be made.

Both models are complementary to each other and the results from each one can be interpreted in a fashion unique to each. The network flow-based model allows for different costs for using a new kit and a reclaimed kit. Whereas the IP model is an

aggregate planning model with respect to the consumption of new and reclaimed kits, i.e. it does not differentiate between the two types of kits for reconfiguration. The solution from the network flow-based model specifies the exact kit inventory to use by means of a path that can be traced by post processors as explained in Chapter 5. The IP formulation ensures the reconfiguration process consumes all necessary kits and a product is reconfigured to a compatible one. Thus the IP formulation does not suffer from any of the feasibility issues identified for the network flow-based model.

This chapter discussed in depth the modeling approaches employed. It has provided the user with an illustrative example, two optimization formulations, and discussed the merits and limitations of each formulation.

CHAPTER 5

DEVELOPMENT OF THE DECISION SUPPORT SYSTEM

This chapter describes the decision support system environment and the interface between the DSS and the optimization software. Section 5.2 explains the inputs to the DSS and the need for post processing the solution to the network flow-based model.

5.1 Decision Support System

As an integral part of this research effort a prototype DSS was developed. This DSS allows for the input of product and kit data. It serves as an interface, where a user can input data such as the bill of materials (BOM) for each product, on-hand inventory and the demand values for kits and products.

The DSS is a Microsoft® Excel / VBA based interface. However, the reconfiguration output will be from the FICO Xpress Optimization Suite Version 7.3 (Heipcke [2012]). The aim of the DSS is to facilitate data entry, input processing, interfacing, post-processing and to provide a familiar interface to the end user.

- Inputs to the DSS: Total number of products for sale, total number of kits for sale, BOM of products, inventory and demand data of products and kits, and cost of reconfiguration.

- Output of the DSS / optimization model: Cost of product reconfiguration, product reconfiguration decision, and the reclaimed kits generated.

A screen shot of the DSS has been provided in Figure 5.2 as a reference. It shows the

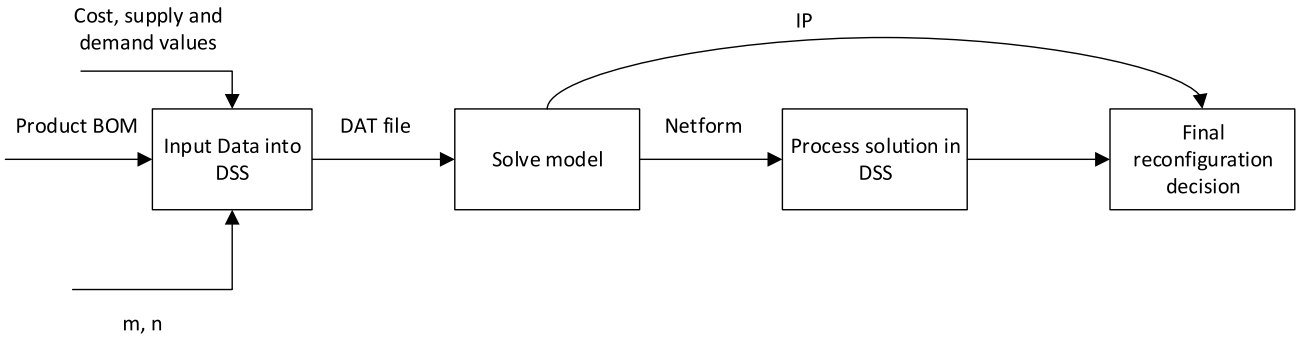

Figure 5.1: Interfacing of Optimization Models and DSS

data for the example described in section 4.1. The inventory and demand values are randomly generated. Macro 1 titled "RUN" must be executed after entering values for m and n. This will prepare and generate an area for entering the BOM, inventory, and demand values. After the data has been entered "Run 2" is executed to begin interfacing with the Xpress models. The other buttons labeled, Clear, Direct Sale, D Ran, Random and Rand inst have been developed for generating test instances.

5.2 Interfacing DSS with the Optimization Software

The DSS has been implemented in Microsoft® Excel / VBA whereas the network flow-based and IP formulation have been implemented in the FICO Xpress Optimization Suite. Since both packages are distinct, an interface was created using a DAT file generated by the DSS. The DAT file is compatible with both DSS and optimization software. Once user opens the DSS and enters the required data as mentioned in Section 5.1, a Macro must be executed to generate the DAT file. This file will act as input to the optimization software. The same file is used for interfacing with the network flow-based and IP formulations.

The DAT file consists of number of products and kits, BOM, constituents of node

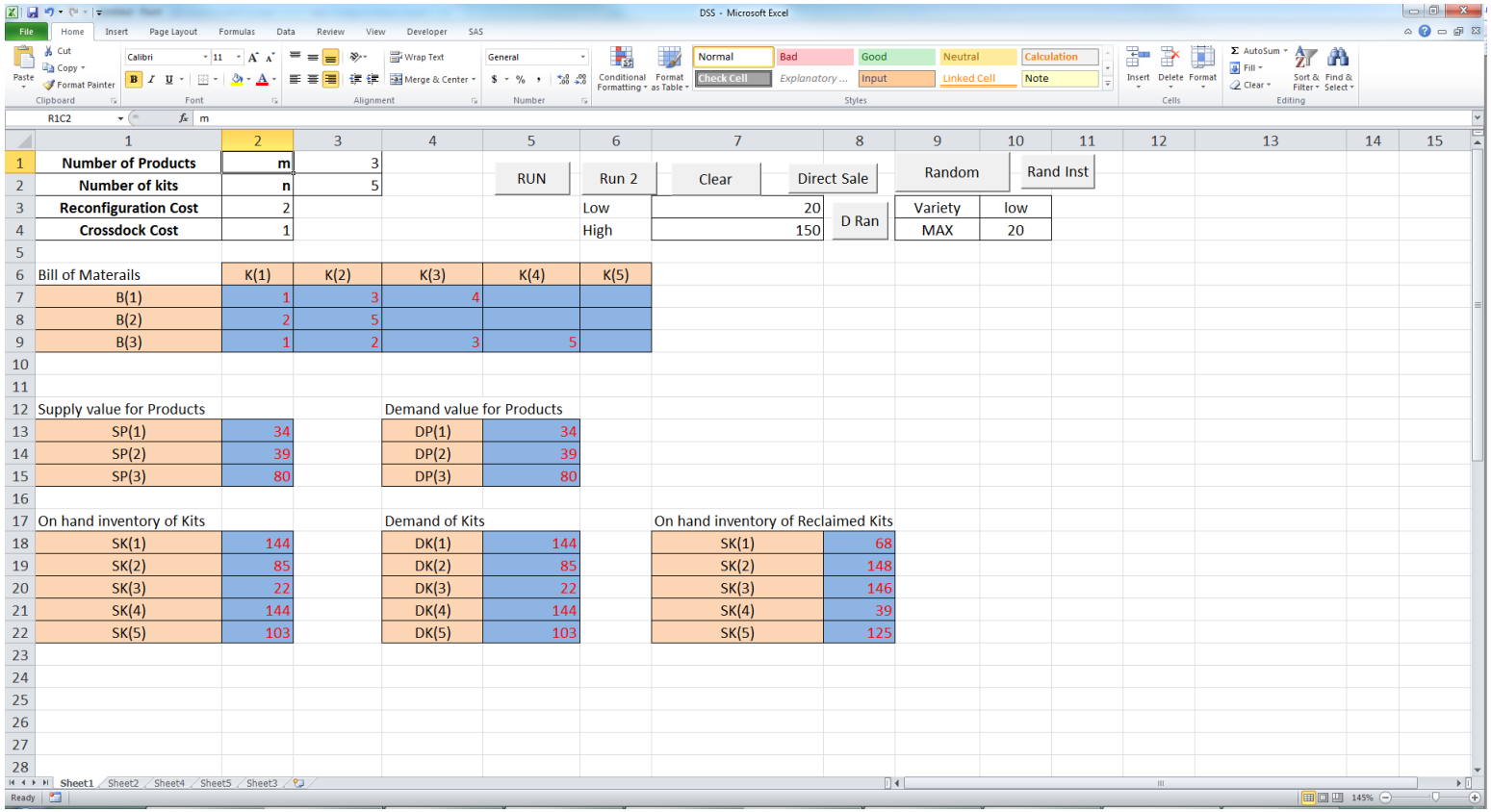

Figure 5.2: Screenshot of DSS

set, supply and demand values for products and kits and the reconfiguration cost and cross dock cost. The use of the DAT for both formulations is explained in subsequent sections. Each implementation of the model is a stand alone *.mos file.

5.2.1 Network Flow-based Implementation

The implementation of the network flow-based formulation constructs the network and solves the model by assigning flows to the arcs. Model solution does not directly state which product consumes which kit but states only flows assigned to each tail-head combination of the arc set. The information with respect to which product consumes which kit needs to be extracted from the solution presented by the optimization model. This information is also essential in recognizing the donor and reconfigured product and quantity of donor item consumed for reconfiguration. This extraction is performed in multiple stages by post-processing solution data. For ease of implementation and display of the final solution, the post processors are MS Excel / VBA based.

Once the Xpress model has been executed it generates two DAT files that re-interface with the DSS for purposes of post-processing. The first model-output file contains number of arcs, number of arcs with assigned positive flows, objective function value, and two counter variables - one, keeping track of reclaimed kits to be generated and second, keeping track of required kits. This file allows the DSS to initialize variables for post-processing.

The second model-output file contains solution to the optimization model - the arc set, arcs with positive flow, flows on the arcs, a list of products compatible with any product i, kits required for reconfiguring product i to product j and the final consumption values of kits, products, reclaimed kits and new kits for reconfiguration. The variables receive their values from this file and are echoed on to a worksheet within the DSS.

Post-processing is required to extract which reclaimed kits are generated from reconfiguration decisions. Apart from this, post processing is also used to detect *invalid paths* that may be used in reconfiguration by the solution to the network flow-based model. An invalid path refers to a case where a compatible product i consumes either fewer than necessary kits or incorrect kits to be reconfigured to product j. This path is undesirable in the final solution since it does not reflect a situation that would occur in the original problem. Generic versions of the post processing algorithms have been provided for the reader to understand how a path is traced, how inventory of reclaimed kits is augmented, and how an invalid path is detected.

Algorithm 1 Post Processing A

%comment: After all post processing input files(output from optimization models) are initialized. Let $Arc(x, y)$ = array of arcs with flow > 0. $T = A(x, 1)$ is Tail/Origin and $H = A(x, 2)$ is Head/Destination of Arc x. X = size of the arc-set. m = total number of products and n = total number of kits %

$H = 0$, $T = 0$

for $i = 1$ to m **do**

    for $j = 1$ to X **do**

        % comment: Find a **supply node** in the arc set, A and store it in T and correspondingly store head of the arc in H. %

        if $A(j, 1) = i$ and $A(j, 2) <> i + m$ **then**

            $T = A(j, 1)$

            Print T on spreadsheet

            $H = A(j, 2)$

            % comment: The head of Arc(j,)will either exist in N_k^R or N_k^G. %

            if $A(j, 2) \in N_k^R$ **then**

                $H = A(j, 2)$

                Print H on the spreadsheet

                Label1:

                % comment: Scan arcs. Find an arc with its Tail being H %

                for $k = 1$ to X **do**

                    % comment: Check if Tail is H and whether head of $Arc(k, 2)$ exists in N_k^R or N_k^G or N_p^- %

if $A(k,1) = H$ and $A(k,2) \in N_k^R$ **then**

$H = A(k,2)$

Print H on the spreadsheet

$k = 1$

% Re-initialize search %

end if

if $A(k,1) = H$ and $A(k,2) \in N_k^G$ **then**

Print H on the spreadsheet

$k = 1$

% Re-initialize search %

end if

if $A(k,1) = H$ and $A(k,2) \in N_p^-$ **then**

if H is compatible with T **then**

Print H on the spreadsheet

$k = 1$

% Re-initialize search %

end if

end if

end for

End Label1:

end if

39: **if** $A(j,2) \in N_k^R$ **then**

40: $H = A(j,2)$

41: Print H on the spreadsheet

42: GOTO Label1

43: **end if**

44: **end if**

45: **end for**

46: **end for**

47: % comment:All the paths are printed on to a single column in the spreadsheet %

Algorithm 2 Post Processing B

%comment: Compare path created by reconfiguration of product i to product j as traced by Post Process A, to augment inventory of reclaimed kits generated by the reconfiguration. Let $R()$ be an array containing all reclaimed kits generated by conversion of any compatible product i to any compatible product j %

for All printed cells **do**

    Find the starting node number $i \in N_p^+$

    node(supply) number = S

    for All remaining cells in the column **do**

        Find the end node number $j \in N_p^-$

        node(demand) number = D

        for All members $\in R()$ **do**

            Find pointer in R where S is located. Store it as $P1$

            Find pointer in R where D is located. Store it as $P2$

            for All elements between $P1$ to $P2$ **do**

                Print all

                %comment: These elements represent the reclaimed kits that are generated for converting j to i. %

                %comment: The minimum flow corresponding to the path traced for j to i in Post Process A will equal the total units of reclaimed kits being generated. In the case of our DSS, the minimum flow is calculated using a Excel function program coupled with input from the second model-output file. %

            end for

        end for

    end for

end for

Algorithm 3 Post Processing C

%comment: This algorithm prints all kits (new and reclaimed) that can be used for reconfiguring i to j based on the path traced by Post Processor A. It also prints the number of kits that should be used for the reconfiguration. Let $Q()$ be an array containing all new and reclaimed kits consumed by conversion of any compatible product i to j %

for All printed cells **do**

    Find the starting node number $i \in N_p^+$

    node(supply) number = S

    for All remaining cells in the column **do**

        Find the end node number $j \in N_p^-$

        node(demand) number = D

    end for

end for

for All members $i \in Q()$ **do**

    Find pointer in Q where S is located. Store it as $P1$

    Count the number of elements of Q that were traversed

    Find pointer in Q where D is located. Store it as $P2$

    if Count exceeds twice the number of kits **then**

        GoTo Label1

        Count = 0

    end if

18: **for** $k = P1 + 1$ to $Count + 1$ **do**

19: **if** $Q(k) \neq D$ **then**

20: Print all elements of array Q next to elements from Post Processor A

21: Print $Count/2$

22: %comment: $Count/2$ refers to the number of kits taking part in the reconfiguration. %

23: **end if**

24: **end for**

25: Label1:

26: **end for**

The post processors are essential to extract reconfiguration information - which donor product is being reconfigured to which product. They also help determine the ending reclaimed kit inventory. Apart from these functions the post processors also detect invalid paths and invalid reconfigurations. If an invalid path / reconfiguration is detected, then with human intervention the solution to the model can be modified to meet the original problem requirements. The post processors aid in this process by displaying the information necessary for validating a reconfiguration. However, there is no guarantee that the modifications needed can be easily identified.

During testing it was discovered that the above mentioned set of post processors were unable to trace all paths resulting in reconfiguration. An approach was developed by means of scanning the solution to the network flow-based formulation; similar to post processor A but beginning from product j (demand side) and ending at product i (supply side). Collectively, the execution of post processors A, B, C, and D, E, F ensures all paths from the solution are traced.

Extra set of post processors C, D, and E were developed to address the above issue. The major difference is the starting point of the scan. Post processors A, B,

and C begin scanning the solution from the supply side of the network i.e. for a reconfiguration of product i to product j, these post processors begin from product i and end at product j. Whereas post processors C, D, and E begin scanning from the demand side i.e. for the aforementioned reconfiguration, C, D, and E begin scanning from product j and end at product i . These extra post processors have ensured that all reconfigurations are correctly traced. Similar to post processors A, B, and C the new ones D, E, and F also print the traced path, generated reclaimed kits, and detected invalid paths on a worksheet. Both sets of post processors work on the same DAT file generated by Xpress hence, post processor set A,B, and C must be disabled before the C, D, E are executed, this merely means disabling the macro for one set of post processors. Both sets of post processors also print on the same worksheet, hence clearing the worksheet must be ensured in order to maintain solution integrity.

5.2.2 Integer Programming based Implementation

The DAT file used to solve IP is same as the one used for the network flow-based formulation. The solution from the Xpress model can be directly used and requires minimal post processing. The IP formulation ensures an optimal solution for all cases and does not encounter any "invalid paths" similar to ones encountered in the network formulation. The post processors involved for the IP are to determine how many units of a particular product are being sold as what product and the final consumption of the products. Products whose demand has not been met completely can also be identified from the model solution. Two distinct cases exist - product i is sold as j or it is sold as is. The number of units consumed by product i equals the number of units of kits consumed $\in \mathcal{B}_j \setminus \mathcal{B}_i$. The number of units of reclaimed kits generated is also equal but $\in \mathcal{B}_i \setminus \mathcal{B}_j$. The model also considers the generation of reclaimed kits and its use in the same time period.

As mentioned in the previous section, the IP is an aggregate planning model and

does not differentiate between new and reclaimed kits for reconfiguration. Hence, the cost of consuming a new kit is same as cost of consuming a reclaimed kit. It has been ensured the direct sale of kits is only met by the new kits.

This chapter discussed the development of the prototype decision support system environment. It provided an overview of the interfacing of both models with the DSS and the post processing algorithms were explained. The chapter also explained the use of DSS for viewing and post processing model solutions.

The post processing algorithm provided the reader with insights on how information can be extracted from the network flow-based solution. Lastly, the chapter also described the DSS and its interfacing with both models.

CHAPTER 6

NUMERICAL EXPERIMENTS

This chapter describes the numerical experiments conducted to test developed models. Section 6.1 provides details of test bed, test data, and conditions under which the models were tested. Section 6.2 describes specific test instances and insights gained from both network flow-based formulation and integer programming formulation.

6.1 Model Test Bed and Test Data

Both the network flow-based formulation and integer programming formulation are new models in the product reconfiguration context. Real world data was difficult to obtain and model testing was limited to randomly generated data. Availability of real world data would have allowed us to compare the output of our models to real world solutions and deliver better insights. The DSS developed as a result of this research effort adds value by allowing users to generate test data and gain insights by executing the models on test instances. The interface of the DSS makes it easier to enter and manipulate data as well. Since the models re-interface with the DSS, the final solution can be viewed in a spreadsheet format. Spreadsheets allow for ease of implementation and data handling. For the purposes of testing, a macro-enabled test bed was created utilizing the VBA environment. Details of the test bed are as follows.

The test bed requires the maximum number of products that will be part of the problem to be solved and the BOM of each product is randomly generated. Three product varieties were considered.

- Low product variety varying from $20 - 40\%$ of maximum number of products.
- Medium product variety varying from $45 - 80\%$ of maximum number of products.
- High product variety varying from $85 - 100\%$ of maximum number of products.

While generating test instances, each product variety was chosen and the number of kits for the problem was limited to 6 types for the low product variety, 7 for medium product variety and 10 for high product variety to. The maximum number of products was fixed at 20. The test bed ensured that the BOM of each product was unique and there was no repetition of kits for a product. Each product consisted of only one unit of any type of kit and the kits were ordered from smallest to largest in the BOM. For every instance of low, medium or high product variety, the BOM of any product can be classified as low, medium, and high density. Density refers to the number of kits that are part of BOM of a product. Hence, for each variety, the number of kits in BOM of any product was varied as follows

- Low density varying from $20 - 40\%$ of maximum number of kits.
- Medium density varying from $41 - 80\%$ of maximum number of kits.
- High density varying from $81 - 100\%$ of maximum number of kits.

For each variety, two instances were generated and tested on the models. The inventory and demand parameters for a product, kits, and reclaimed kits were also randomly generated. The lower and upper bound of inventory and demand values were specified as 20 and 150 respectively. For this study randomly generated data refers to data generated by using the MS Excel randbetween function with lower and upper bound values. The reconfiguration cost was set to be 2 units per consumption of reclaimed kit, and cross dock cost was unity.

Each model was first tested for direct sale requests of products and kits and this was treated as the base case (macro generated equal inventory and demand values). For a direct sale request the inventory values equal exactly the demand values. Once it was established all direct sale requests are met and no product or kit demand goes unfulfilled; the consumption of reclaimed kits was checked (network flow-based model) to ensure that reclaimed kits are not consumed to fulfill direct sale. Once the base case was tested, reconfiguration requests were tested by removing equality on inventory and demand values.

6.2 Results and Discussion

The model was tested for 2 instances of each product variety, resulting in a total of 6 instances. The results were manually checked for consistency. Since the IP has a single cost for consumption of new and reclaimed kits, the objective function values cannot be directly used for the comparison of solutions from the two formulations. We chose to compare the reconfiguration decisions generated from both the models. This meant manually checking which product is being reconfigured to which product. This process though manual, ensured that the comparisons of models were "fair". The network flow-based model requires more effort to ensure consistency with regards to consumption of kits, generation of reclaimed kits, and validity of paths. All the post processors were utilized for this purpose.

In case of IP, consumed kits and generated reclaimed kits were checked against the ones being consumed and released for a reconfiguration of product i to j. The compatibility of both i and j was also verified.

6.2.1 Low Product Variety

First instance generated 5 products and 6 kits with direct sale requests.All demand

was met and unfulfilled demand was reported correctly.

The second instance had 6 products and 6 kits. One invalid path was detected. Invalid path was corrected by consuming necessary kits and hence the solution was made feasible to the original problem. All other reconfigurations were as expected and were directly comparable with the IP solution. The augmentation of inventory of reclaimed kits was also as expected.

For the instance with only direct sale requests, both models act exactly alike. The unfulfilled demand also remains the same and solution can be verified. In the second instance, due to the inherent nature of the network flow-based formulation an invalid path was created, allowing a product to consume one less kit than necessary and be converted to a compatible product.

6.2.2 Medium Product Variety

As the size of the model increased, the difficulty in verifying results and gaining insights also increased. The first instance generated 12 products and 7 kits. This instance consisted of 4 high density, 2 medium density, and 6 low density products.

Analysis of the output from the network flow-based model showed a generation a total of 7 reconfigurations and both sets of post processors were able to trace 7 reconfigurations. Further analysis of the post processor results showed that 2 invalid paths existed. These paths were successfully detected. One invalid path consumed fewer than the required number of kits and the other consumed kits that were not required for the reconfiguration. It was also noticed that the post processors were unable to trace a path for the following condition - the path consisted of product 1 consuming 62 units of reclaimed kit 2 and being sold as 57 units of product 2, and remaining 5 units as product 10. Product 7 consumed 46 units of reclaimed kit 2 to be sold as 46 units of product 10. The post processors traced a correct path of reconfiguring product 1 to product 2, but an incorrect path where product 7 is

reconfigured to product 2. Since reconfiguration path of product 7 to product 10 and product 1 to product 2 shared common kit 2, the trace of reconfiguration of product 7 discovered product 2 and not 10. This incorrect path is traced because the post processor finds compatible product i, j by choosing the lowest subscript first.

Solution to the IP model states a total of 8 reconfigurations of which 4 are common with output from network flow-based model. Both the models also consume all available product supply to meet all possible product demand, the unfulfilled demand of kits was also same, but only after the removal of invalid paths generated by solution to the network flow-based model.

Second instance had 9 products and 7 kits. This instance consisted of 2 high density, 3 medium density, and 4 low density products. Analysis of the solution to the network flow-based model showed total of 5 reconfigurations, of which the post processor detected 3 as incorrect. Further analysis showed that product 4 was reconfigured to product 6 with which it is incompatible. Secondly, the reconfiguration of product 9 to 3 consumed incorrect kits and reconfiguration of product 9 to 8 consumed one kit fewer than necessary for a complete and valid reconfiguration.

Solution to IP generated 5 reconfigurations as well, of which two are common with the network flow-based solution, the invalid reconfigurations are absent in this solution. Both solutions also exhaust all product supply to meet as much demand as possible.

6.2.3 High Product Variety

The first instance had 19 products and 10 kits. This instance consisted of 8 high density, 3 medium density, and 8 low density products.

Solution to the network flow-based model generated a total of 8 reconfigurations out of which only one was analyzed to be correct. All others consisted of invalid paths, by using fewer than the required number of kits for reconfiguration. All product

demand was met by direct sale or by means of reconfiguration. For all kits that were consumed for reconfiguration, at least one was necessary for reconfiguration. Both sets of post processors were able to detect all the reconfigurations that were present in the network model.

Solution to the IP generated 12 reconfigurations of which only one is common with the network flow-based solution. All product demand was met by reconfiguration or direct sale. A unique reconfiguration solution existed, where, 22 units of product 11 were sold as product 14 which had a demand of 22 units and inventory value of 20 units. This reconfiguration resulted generated 22 units of $\{3, 4, 7\}$ reclaimed kits, all of which were used in other reconfigurations. The validity of this reconfiguration is described in section 6.3.

The second instance had 17 products and 10 kits. This instance consisted of 2 high density, 2 medium density, and 13 low density products.

Solution from network model generated 9 reconfigurations. Post processing revealed that three were invalid. One product was reconfigured to an incompatible product. Two reconfigurations consisted of invalid paths. Both used one kit less than that was necessary for reconfiguration. All product supply was consumed yet the demand for 6 products was unfulfilled; and demand for 3 types of kits was unfulfilled.

Solution from IP generated 9 reconfigurations of which 3 were common with the network flow-based model. The IP also exhausted all product supply to meet as much demand as possible.

The analysis of solutions from the instances makes it clear that both models work as expected and the post processors effectively detect - invalid paths and invalid reconfigurations. It is to be noted that the augmentation of reclaimed kits is not reported. As the IP model can consume reclaimed kits in the periods they are released, direct comparison would not be possible. For the network flow-based model the generation of reclaimed kits can be computed manually after post processing.

For all instances there was available inventory of reclaimed kits that could be used to meet the reconfiguration requests. Section 6.3 also contains an instance where a new kit is consumed to meet reconfiguration request over direct sale. The BOM, supply and demand values for all instances can be found in Appendix B.

The following tables summarize the test data, number of reconfigurations, detection of invalid paths from network flow-based model, common / overlapping reconfigurations from both network flow-based model and IP model. Overlapping reconfigurations are the reconfiguration decisions that are common between both types of models. Table 6.1 provides a summary of the data while table 6.2 describes the reconfigurations detected.

Table 6.1: Test Data

Product Variety	Instance	Number of products (m)	Number of kits (n)	High density	Medium Density	Low Density
Low	1	5	6	0	3	2
	2	6	6	2	2	2
Medium	1	12	7	4	2	6
	2	9	7	2	3	4
High	1	19	10	8	3	8
	2	17	10	2	2	13

Table 6.2: Comparison of Reconfigurations from Netform and IP models

Product Variety	Instance	Number of reconfigurations	Overlapping reconfigurations	Invalid Path / Reconfigurations
Low	1	NA	NA	NA
	2	Netform = 2, IP = 3	1	Netform = 1, IP = 0
Medium	1	Netform = 7, IP = 8	4	Netform = 2, IP = 0
	2	Netform = 5, IP = 5	2	Netform = 3, IP = 0
High	1	Netform = 8, IP = 12	1	Netform = 7, IP = 0
	2	Netform = 10, IP = 9	3	Netform = 3, IP = 0

6.3 Analysis of a Special Case

The importance of considering generation of reclaimed kits and its use within a time period can be understood by analyzing this case. The IP was modified by dropping the $\sum_{\substack{i,j\in\mathcal{P}|(\mathcal{B}_i\cap\mathcal{B}_j\neq\emptyset,\\ k\in\mathcal{B}_i\setminus\mathcal{B}_j,i\neq j)}} X_{ij}$ term from constraints 4.9 and 4.10. Constraint 4.9 is replaced by

$$\sum_{\substack{i,j\in\mathcal{P}|(\mathcal{B}_i\cap\mathcal{B}_j\neq\emptyset,\\ k\in\mathcal{B}_j\setminus\mathcal{B}_i,i\neq j)}} X_{ij} \leq S_k + S'_k \qquad \forall k \in \mathcal{K}$$

and constraint 4.10 is replaced by

$$\sum_{\substack{i,j\in\mathcal{P}|(\mathcal{B}_i\cap\mathcal{B}_j\neq\emptyset,\\ k\in\mathcal{B}_j\setminus\mathcal{B}_i,i\neq j)}} X_{ij} - S'_k \leq Z_k \qquad \forall k \in \mathcal{K}$$

The rest of the formulation remains the same. This ensured that there is no generation of reclaimed kits and their consumption in the same time period. The data from section 6.2.3 was used for testing this instance. The solution to this IP model was elimination of the unique solution.

The solution to this model states, 20 units of product 14 be sold as direct sale and 2 units of product 11 were reconfigured to meet the remaining demand for product 14. As compared to reconfiguring 22 units of product 11 which yields in 22 units of kit 3 which can be used for other reconfigurations.

Extra units of new kit 3 were used in other reconfigurations which could have been made available from reconfiguring product 11 to product 14 leading to 65 units of kit 3 being unfulfilled as compared to 47 units from the previous solution.

This also shows that a higher priority is placed on consuming a new kit for product reconfiguration over a direct sale. This illustration shows two things - the importance

of considering generation and consumption of reclaimed kits and the priority of consuming a new kit for product reconfiguration over direct sale.

6.4 Importance of Decision Support System

The DSS has proved to be essential in the process of testing and post processing the data. The spreadsheet nature of the DSS has allowed for ease in data handling and made it easy to view the results. The VBA environment has also allowed us to automate the generation of input files to both models. MS Excel randbetween function also aided in the generation of test instances. For the end user, the DSS acts a simple interface that facilitates data entry of BOM, inventory, and demand parameters, execution of models and the visualization of final reconfiguration solution.

This chapter discussed the testing of models and results of test instances. It also explained the importance of the DSS and its application in this research effort.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this chapter we first summarize the research effort, state the findings from the numerical experiments in the conclusions section and then go on to discuss potential avenues for future research.

7.1 Research Summary

The efforts of this research were focussed on recognizing, understanding, and modeling the product reconfiguration process. The product reconfiguration process was modeled by a network flow-based formulation and an integer programming formulation. The network flow-based formulation was desirable because of its linear nature, ease of understanding, and computational ease. Although elegant, it was noticed that the solution to the network formulation allowed flows to exist on paths which were not acceptable as a solution to the original problem being studied. This prompted the development of an IP based formulation. The IP formulation is an aggregate planning model and does not represent a detailed solution similar to the network flow-based model. However, the IP formulation ensures optimality to the original problem.

As enterprises aim for mass customization, the product reconfiguration strategy allows the enterprise to cater to a larger base of customers. The stocking of products at a DC and modifying the structure as demand is realized also reduces backorders to some extent. This will improve the service level for end customers as well. Inventory that has been made obsolete by introduction of new products can be sold to end customers by upgrading them to the highest available option. The product reconfig-

uration strategy has many advantages and our research effort has added value to this field by providing a classification of the supply chain strategy, developing a network flow-based formulation, IP formulation, and by identifying the merits and demerits of both formulations.

We also developed a prototype DSS that facilitates data entry, storage, handling, processing and visualization of reconfiguration decisions. The VBA environment has aided in post processing and automating data handling.

7.2 Conclusions

The specific conclusions from the research effort are as follows.

- For smaller problems with number of products and kits less than 5, the network flow-based model is an attractive modeling structure that makes it easy to understand and explain reconfiguration solutions.

- The IP formulation provides accurate solutions for larger problems but the different costs of consuming a reclaimed kit and a new kit have not been modeled.

- For all solutions to the network model a set of post processors are necessary. These post processors are essential to overcome the inherent limitations of a network model generating invalid reconfigurations. The solution might allow a product to be reconfigured to an incompatible one and or allow reconfiguration by consuming incorrect kits or fewer than required number kits. These limitations also cause difficulty in developing the final number of reclaimed kits generated. However, the elegance of the network model cannot be ignored as the solution clearly states whether a new kit or a reclaimed kit has been used in a reconfiguration.

- The IP reconfiguration solutions can be difficult to explain sometimes as the IP (as it is supposed to) finds solutions that may not be even considered in

a manual system. For example, not meeting a product demand by direct sale but consuming another product with excess inventory and using the generated reclaimed kits for other reconfiguration to meet more product demand. This unusual situation arises because of the inclusion of the generation of reclaimed kits. Using reclaimed kits for other reconfigurations in the manual decision making process is not straightforward and can be overlooked. This solution is better because reclaimed kits are released and consumed in the same period instead of being left over for consideration in the next time period. This highlights the importance and need for a math model in the decision making process. A specific case has been covered in section 6.2.3.

- The computational times for both models were very short, even the largest problem was solved under a second; since human intervention is necessary to make sense of the solution, extra time is required for post processing the solution.

7.3 Avenues for Future Work

In this research effort we have focussed on understanding, documenting and modeling the product reconfiguration process. The insights gained from both network and integer programming models have been limited by the non-availability of real world data. If real world data was available the output of the models could have been compared with the solution developed by an operations manager.

1. The current network model solution is arc flow-based. The following avenues can be explored for the network model solution in future.

    - Develop a path based formulation for the network model. This will eliminate the inclusion of invalid flows or paths and also reconfiguration involving two incompatible products.

- Develop approach to re-solve the network flow-based model solution in case any invalid path / reconfiguration are detected.

2. The IP formulation has a single cost for consuming kits that is based on the number of kits consumed in the reconfiguration. It also does not distinguish between the consumption of new and reclaimed kits.

   - Developing a formulation where costs of consuming a new kit and reclaimed kit are different. This will allow the model to mimic the real world situation more closely and the results can be more directly applicable.
   - Modeling the generation of ending reclaimed kit inventory.

3. For both the network based formulation and the IP cost of removal of kits can be taken into consideration. This procedure could improve the solution by placing a priority on a product reconfiguration that involves fewer number of kits being removed and added, as compared to one where only addition of kits is considered. Currently both models do not consider reconfiguring a product j whose BOM is a subset of product i, by adding a cost to remove kits this reconfiguration will be included in the optimal solution hence enriching the solution space. This addition may also change the unique solution obtained in section 6.2.3.

4. From the perspective of the DSS, following avenues can be explored

   - Developing VBA code to directly access and execute models in Xpress and hence utilizing both packages to perform computation, data handling and thus improving solution visualization in MS Excel environment.

5. Allowing for BOM to consist of multiple units of a single kit, this will require equivalent changes to the network model and the IP formulation.

This research effort has many avenues that can be explored and a few have been documented as a part of this research effort. The effect on the supply chain, demand forecast evolution, availability of multiple reconfiguration centers, modeling the actual reconfiguration process are some of the other avenues that come to mind and should be explored.

BIBLIOGRAPHY

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1st edition.

Aitken, J., Childerhouse, P., and Towill, D. (2003). The impact of product life cycle on supply chain strategy. *International Journal of Production Economics*, 85(2):127–140.

Davis, T. (1993). Effective supply chain management. *Sloan management review*, 34:35–35.

DeHoratius, N. and Raman, A. (2008). Inventory record inaccuracy: An empirical analysis. *Management Science*, 54(4):627–641.

Heipcke, S. (2012). Xpress–mosel. In *Algebraic Modeling Systems*, pages 77–110. Springer.

Jensen, P. A. and Bard, J. F. (2003). *Operations Research: Models and Methods*, volume 1. John Wiley & Sons Inc.

Lambert, D. M. (2008). *Supply Chain Management: Processes, Partnerships, Performance*. Supply Chain Management Inst.

Lee, H. L. (2002). Aligning supply chain strategies with product uncertainties. *California Management Review*, 44(3):105–119.

Pagh, J. D. and Cooper, M. C. (1998). Supply chain postponement and speculation strategies: How to choose the right strategy. *Journal of Business Logistics*, 19:13–34.

Rinehart, R. F. (1960). Effects and causes of discrepancies in supply operations. *Operations Research*, 8(4):543–564.

Sabin, D. and Weigel, R. (1998). Product configuration frameworks-a survey. *Intelligent Systems and their Applications, IEEE*, 13(4):42–49.

Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E. (2004). *Managing the Supply Chain: The Definitive Guide for the Business Professional*. McGraw-Hill Companies.

Swaminathan, J. M. and Lee, H. L. (2003). Design for postponement. *Handbooks in Operations Research and Management Science*, 11:199–228.

Van der Laan, E. and Salomon, M. (1997). Production planning and inventory control with remanufacturing and disposal. *European Journal of Operational Research*, 102(2):264–278.

Van der Laan, E., Salomon, M., and Dekker, R. (1999). An investigation of lead-time effects in manufacturing/remanufacturing systems under simple push and pull control strategies. *European Journal of Operational Research*, 115(1):195–214.

Van Hoek, R. I. (2001). The rediscovery of postponement: A literature review and directions for research. *Journal of Operations Management*, 19(2):161–184.

Venkatesh, S. and Swaminathan, J. M. (2004). Managing product variety through postponement: Concept and applications. In *The Practice of Supply Chain Management: Where Theory and Application Converge*, pages 139–155. Springer.

APPENDIX A

Notated Three-Product Example:

$m = 3$ since there are three products and $n = 5$ for a total of five kits.

1. Set of products and kits:

   - $\mathcal{P} = \{p_1, p_2, p_3\}$

   - $\mathcal{K} = \{k_1, k_2, k_3, k_4, k_5\}$

2. Bill of Materials:

   - $\mathcal{B}(p_1) = \{k_1, k_4, k_4\}$

   - $\mathcal{B}(p_2) = \{k_2, k_5\}$

   - $\mathcal{B}(p_3) = \{k_1, k_2, k_3, k_5\}$

3. Set of Nodes

$$N = N_p^+ \cup N_k^+ \cup N_p^- \cup N_k^- \cup N_k^R \cup N_k^C \cup N_k^G$$

   - $N_p^+ = \{p_1^{(1)}, p_2^{(1)}, p_3^{(1)}\}$

- $N_k^+ = \{k_1^{(1)}, k_2^{(1)}, k_3^{(1)}, k_4^{(1)}, k_5^{(1)}\}$

- $N_p^- = \{p_1^{(2)}, p_2^{(2)}, p_3^{(2)}\}$

- $N_k^- = \{k_1^{(2)}, k_2^{(2)}, k_3^{(2)}, k_4^{(2)}, k_5^{(2)}\}$

- $N_k^R = \{k_1^{(3)}, k_2^{(3)}, k_3^{(3)}, , k_4^{(3)}, k_5^{(3)}\}$

- $N_k^C = \{k_1^{(4)}, k_2^{(4)}, k_3^{(4)}, , k_4^{(4)}, k_5^{(4)}\}$

- $N_k^G = \{k_1^{(5)}, k_2^{(5)}, k_3^{(5)}, , k_4^{(5)}, k_5^{(5)}\}$

4. Set of Arcs

   - $\mathcal{B}(p_3^{(2)}) \setminus \mathcal{B}(p_1^{(1)}) = \{k_{v2}, k_{v5}\}$

     In this case $r = 2$.

   - $\mathcal{B}(p_3^{(2)}) \setminus \mathcal{B}(p_2^{(1)}) = \{k_{v1}, k_{v3}\}$

     In this case $r = 2$.

   - $\mathcal{B}(p_1^{(2)}) \setminus \mathcal{B}(p_3^{(1)}) = \{k_{v4}\}$

     In this case $r = 1$.

$A = \Big\{(k_1^{(1)}, k_1^{(4)}), (k_1^{(4)}, k_1^{(2)}),$

$(k_2^{(1)}, k_2^{(4)}), (k_2^{(4)}, k_2^{(2)}),$

$(k_3^{(1)}, k_3^{(4)}), (k_3^{(4)}, k_3^{(2)}),$

$(k_4^{(1)}, k_4^{(4)}), (k_4^{(4)}, k_4^{(2)}),$

$k_5^{(1)}, k_5^{(4)}), (k_5^{(4)}, k_5^{(2)}),$

$(p_1^{(1)}, p_1^{(2)}), (p_2^{(1)}, p_2^{(2)}), (p_3^{(1)}, p_3^{(2)}), (p_3^{(1)}, p_2^{(2)}),$

$(p_1^{(1)}, k_2^{(3)}), (k_2^{(3)}, k_5^{(3)}), (k_5^{(3)}, p_3^{(2)}),$

$(p_1^{(1)}, k_2^{(5)}), (k_2^{(5)}, k_5^{(5)}), (k_5^{(5)}, p_3^{(2)}),$

$(k_2^{(5)}, k_5^{(3)}), (k_2^{(3)}, k_5^{(5)}),$

$(p_2^{(1)}, k_1^{(3)}), (k_1^{(3)}, k_3^{(3)}), (k_3^{(3)}, p_3^{(2)}),$

$(p_2^{(1)}, k_1^{(5)}), (k_1^{(5)}, k_3^{(5)}), (k_3^{(5)}, p_3^{(2)}),$

$(k_1^{(3)}, k_3^{(5)}), (k_1^{(5)}, k_3^{(3)}),$

$(p_3^{(1)}, k_4^{(3)}), (k_4^{(3)}, p_1^{(2)}),$

$(p_3^{(1)}, k_4^{(5)}), (k_4^{(5)}, p_1^{(2)}) \Big\}$

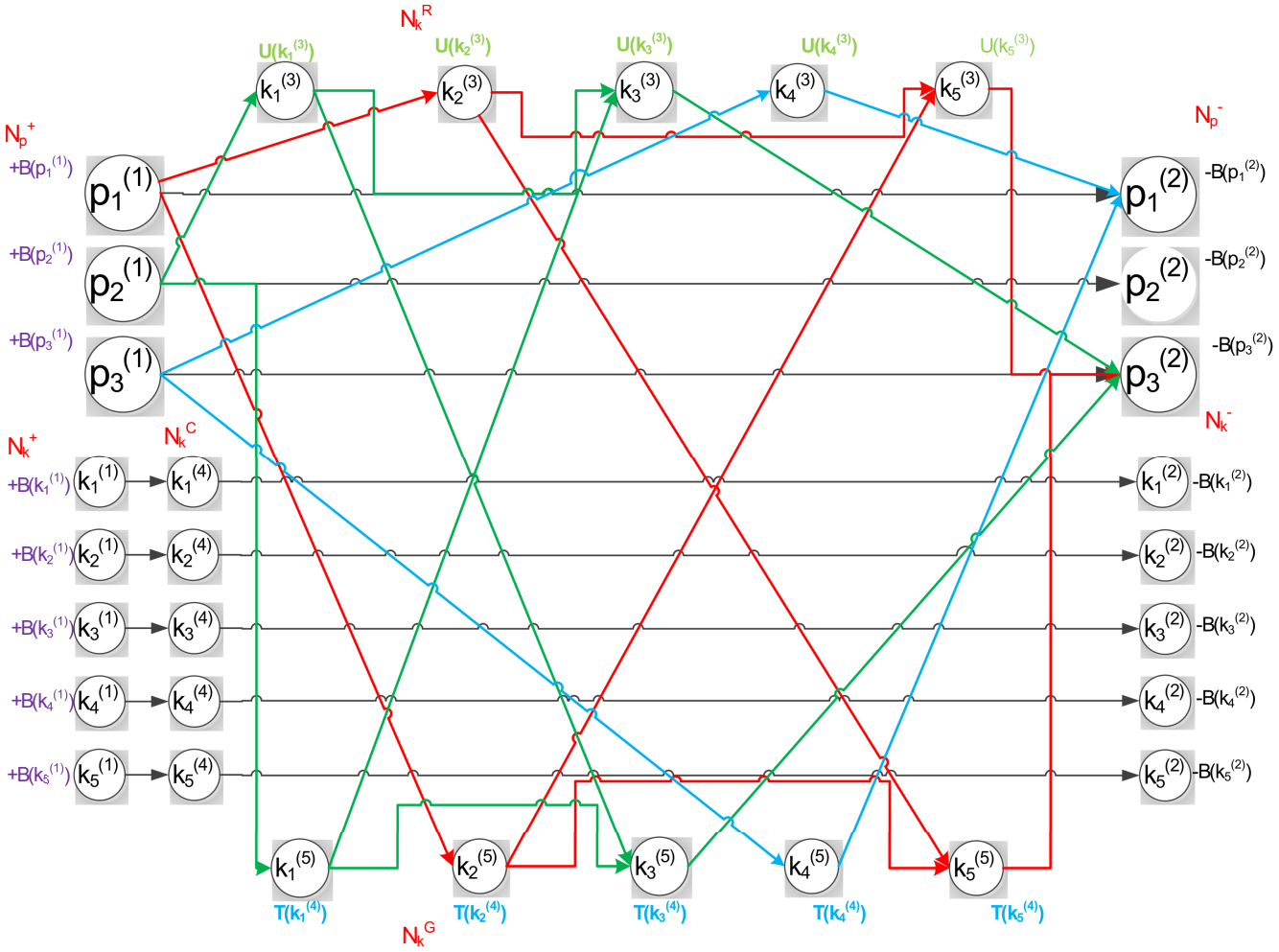

Figure A.1: Network Flow-based Model Illustration of Three-Product Example

APPENDIX B

The Appendix contains test data from chapter 6. The screen shots of DSS show the Bill of material for m products, containing maximum of n kits and randomly generated values for inventory and demand of products and kits with a $low = 20$ and $high = 150$. The maximum number of products is limited to 20.

The *correct consumption* column for each table depicts the kits that *Donor Product* i can consume to be reconfigured to *Product j*. The donor product has to consume exactly half the number of kits from the *correct consumption* column. The first half denotes the reclaimed kits and second half denotes new kits. The numbers denote node labels for the solution to network flow-based formulation. The following example describes how to read the table.

Consider reconfiguration of product 9 to product 8 from table B.6. This is an invalid reconfiguration as detected by post processors 1 and 2. The correct reconfiguration would consume exactly 3 kits from $\{34, 35, 36, 48, 49, 50\}$ the first three elements of the set denote reclaimed kits and the remaining three denote new kits. This set represents set of reclaimed kits $\{2, 3, 4\}$ and new kits $\{2, 3, 4\}$ as seen from the set of N_k^R and N_k^G. A correct consumption can consume any combination of the set by ensuring that a kit is consumed only once. Hence, a consumption of $\{34, 48, 49\}$ will be incorrect. Since 34 and 48 both represent kit 1 (reclaimed and new respectively)

Each table is preceded by the set of N_k^R and N_k^G representing set of reclaimed kits and set of new kits respectively.

B.1 Low Product Variety

B.1.1 Test Instance 1

Table B.1: Low Product Variety Test Instance 1 Network Flow-based and Integer Programming Model Output

Donor Product i	Reconfigured Product j	Units	Consuming kits	Releasing	Valid	Trace Origin	Correct Consumption
1	1	20	NA	NA	Yes	NA	NA
2	2	117	NA	NA	Yes	NA	NA
3	3	32	NA	NA	Yes	NA	NA
4	4	46	NA	NA	Yes	NA	NA
5	5	105	NA	NA	Yes	NA	NA

B.1.2 Test Instance 2

$$N_k^R = \{25, 26, 27, 28, 29, 30\},\ N_k^G = \{37, 38, 39, 40, 41, 42\}$$

Table B.2: Low Product Variety Test Instance 2 Network Flow-based Programming Model Output

Donor Product i	Reconfigured Product j	Units	Consuming kits	Releasing	Valid	Trace Origin	Correct Consumption
3	4	21	$\{26, 30\}$	$\emptyset$	No	PP1	$\{26, 29, 30, 38, 41, 42\}$
6	4	3	$\{29\}$	$\emptyset$	Yes	PP1	$\{29, 41\}$
3	4	21	$\{26, 30\}$	$\emptyset$	No	PP2	$\{26, 29, 30, 38, 41, 42\}$
6	4	3	$\{29\}$	$\emptyset$	Yes	PP2	$\{29, 41\}$
1	1	41	NA	NA	Yes	NA	NA
2	2	72	NA	NA	Yes	NA	NA
3	3	98	NA	NA	Yes	NA	NA
4	4	22	NA	NA	Yes	NA	NA
5	5	20	NA	NA	Yes	NA	NA
6	6	49	NA	NA	Yes	NA	NA

Number of Products	**m**	5
Number of kits	**n**	6
Reconfiguration Cost	2	
Crossdock Cost	1	

Supply / Demand	
Low	20
High	150

Product Variety	low
Max Products	20

Bill of Materails	K(1)	K(2)	K(3)	K(4)	K(5)	K(6)	DENSITY	NUMBER OF KITS
B(1)	3	6					**Low dense**	**2**
B(2)	1	3	6				**Med dense**	**3**
B(3)	2	3	4	6			**Med dense**	**4**
B(4)	1	2	4	5			**Med dense**	**4**
B(5)	6						**Low dense**	**1**

On hand inventory of Products

SP(1)	136
SP(2)	141
SP(3)	104
SP(4)	134
SP(5)	122

Demand value for Products

DP(1)	20
DP(2)	117
DP(3)	32
DP(4)	46
DP(5)	105

On hand inventory of Kits

SK(1)	137
SK(2)	94
SK(3)	82
SK(4)	81
SK(5)	144
SK(6)	122

Demand of Kits

DK(1)	121
DK(2)	27
DK(3)	103
DK(4)	106
DK(5)	93
DK(6)	79

On hand inventory of Reclaimed Kits

SK(1)	51
SK(2)	113
SK(3)	78
SK(4)	52
SK(5)	54
SK(6)	142

Figure B.1: Bill of Materials, Inventory and Demand Data for Low Product Variety Test Instance 1

Number of Products	**m**	6
Number of kits	**n**	6
Reconfiguration Cost	2	
Crossdock Cost	1	

Supply / Demand	
Low	20
High	150

Product Variety	low
Max Products	20

Bill of Materails	K(1)	K(2)	K(3)	K(4)	K(5)	K(6)	DENSITY	NUMBER OF KITS
B(1)	1	3					**Low dense**	**2**
B(2)	1	2	4	5			**Med dense**	**4**
B(3)	1	3	4				**Med dense**	**3**
B(4)	1	2	3	4	5	6	**High dense**	**6**
B(5)	5						**Low dense**	**1**
B(6)	1	2	3	4	6		**High dense**	**5**

On hand inventory of Products

SP(1)	107
SP(2)	103
SP(3)	130
SP(4)	22
SP(5)	64
SP(6)	52

Demand value for Products

DP(1)	41
DP(2)	72
DP(3)	98
DP(4)	46
DP(5)	20
DP(6)	49

On hand inventory of Kits

SK(1)	78
SK(2)	139
SK(3)	26
SK(4)	66
SK(5)	120
SK(6)	83

Demand of Kits

DK(1)	95
DK(2)	79
DK(3)	48
DK(4)	146
DK(5)	100
DK(6)	40

On hand inventory of Reclaimed Kits

SK(1)	106
SK(2)	66
SK(3)	106
SK(4)	37
SK(5)	145
SK(6)	136

Figure B.2: Bill of Materials, Inventory and Demand Data for Low Product Variety Test Instance 2

Table B.3 reports the solution from the IP model. 21 and 25 units of kit 3 and kit 4 respectively is unfulfilled.

Table B.3: Low Product Variety Test Instance 2 Integer Programming Model Output

Donor Product i	**Reconfigured Product j**	**Units**	**Consuming kits**	**Releasing**
2	4	21	$\{3, 6\}$	$\emptyset$
6	4	3	$\{5\}$	$\emptyset$
1	1	41	NA	NA
2	2	72	NA	NA
3	3	98	NA	NA
4	4	22	NA	NA
5	5	20	NA	NA
6	6	49	NA	NA

B.2 Medium Product Variety

B.2.1 Test Instance 1

$N_k^R = \{39, 40, 41, 42, 43, 44, 45\}$, $N_k^G = \{59, 54, 55, 56, 57, 58, 59\}$

Table B.4: Medium Product Variety Test Instance 1 Network Flow-based Programming Model Output

Donor Product i	Reconfigured Product j	Units	Consuming kits	Releasing	Valid	Trace Origin	Correct Consumption
1	2	57	{40}	{39, 41}	Yes	PP1	{40, 54}
3	10	30	{44}	{39, 41, 42, 45}	Yes	PP1	{44, 58}
3	8	15	{57, 45}	{39, 40, 41, 42}	No	PP1	{44, 58}
6	8	23	{45}	{39, 40, 41, 42}	Yes	PP1	{45, 59}
7	2	46	{40}	{41}	No	PP1	{40, 42, 44, 45, 54, 56, 58, 59}
1	2	57	{40}	{39, 41}	Yes	PP2	{40, 54}
6	8	23	{45}	{39, 40, 41, 42}	Yes	PP2	{45, 59}
3	9	5	{57}	{40, 42}	Yes	PP2	{57, 43}
3	10	30	{44}	{39, 41, 42, 45}	Yes	PP2	{44, 58}
1	10	51	{40}	{1, 3, 4, 5, 7}	Yes	PP2	{40, 54}
1	1	70	NA	NA	NA	NA	NA
2	2	21	NA	NA	NA	NA	NA
3	3	75	NA	NA	NA	NA	NA
4	4	36	NA	NA	NA	NA	NA
5	5	71	NA	NA	NA	NA	NA
6	6	119	NA	NA	NA	NA	NA
7	7	20	NA	NA	NA	NA	NA
8	8	41	NA	NA	NA	NA	NA
9	9	78	NA	NA	NA	NA	NA
10	10	45	NA	NA	NA	NA	NA
11	11	23	NA	NA	NA	NA	NA
12	12	23	NA	NA	NA	NA	NA

Number of Products	**m**	12
Number of kits	**n**	7
Reconfiguration Cost	2	
Crossdock Cost	1	

Supply / Demand	
Low	20
High	150

Product Variety	medium
Max Products	20

Bill of Materails	K(1)	K(2)	K(3)	K(4)	K(5)	K(6)	K(7)	DENSITY	NUMBER OF KITS
B(1)	1	3	4	5	6	7		**High dense**	**6**
B(2)	2	4	5	6	7			**Med dense**	**5**
B(3)	1	2	3	4	7			**High dense**	**5**
B(4)	1	2	3	4	5	6	7	**High dense**	**7**
B(5)	5	6						**Low dense**	**2**
B(6)	1	2	3	4	6			**High dense**	**5**
B(7)	3	5						**Low dense**	**2**
B(8)	6	7						**Low dense**	**2**
B(9)	1	3	5	7				**Med dense**	**4**
B(10)	2	6						**Low dense**	**2**
B(11)	2	3						**Low dense**	**2**
B(12)	5							**Low dense**	**1**

Figure B.3: Bill of Materials for Medium Product Variety Test Instance 1

On hand inventory of Products

SP(1)	132
SP(2)	21
SP(3)	125
SP(4)	36
SP(5)	71
SP(6)	142
SP(7)	66
SP(8)	41
SP(9)	78
SP(10)	45
SP(11)	23
SP(12)	23

Demand value for Products

DP(1)	70
DP(2)	137
DP(3)	75
DP(4)	77
DP(5)	93
DP(6)	119
DP(7)	20
DP(8)	105
DP(9)	83
DP(10)	126
DP(11)	92
DP(12)	37

On hand inventory of Kits

SK(1)	83
SK(2)	98
SK(3)	30
SK(4)	140
SK(5)	87
SK(6)	73
SK(7)	148

Demand of Kits

DK(1)	37
DK(2)	71
DK(3)	47
DK(4)	58
DK(5)	92
DK(6)	142
DK(7)	43

On hand inventory of Reclaimed Kits

SK(1)	89
SK(2)	125
SK(3)	109
SK(4)	87
SK(5)	103
SK(6)	30
SK(7)	62

Figure B.4: Inventory and Demand Data for Medium Product Variety Test Instance 1

Table B.5: Medium Product Variety Test Instance 1 Integer Program based Programming Model Output

Donor Product i	**Reconfigured Product j**	**Units**	**Consuming kits**	**Releasing**
1	4	11	$\{2\}$	$\emptyset$
1	10	36	$\{2\}$	$\{1,3,4,5,7\}$
1	11	15	$\{2\}$	$\{1,4,5,6,7\}$
3	9	5	$\{5\}$	$\{2,4\}$
3	10	45	$\{6\}$	$\{1,3,4,7\}$
6	5	22	$\{5\}$	$\{1,2,3,4\}$
6	8	1	$\{7\}$	$\{1,2,3,4\}$
7	11	46	$\{2\}$	$\{5\}$
1	1	70	NA	NA
2	2	21	NA	NA
3	3	75	NA	NA
4	4	36	NA	NA
5	5	71	NA	NA
6	6	22	NA	NA
7	7	20	NA	NA
8	8	41	NA	NA
9	9	78	NA	NA
10	10	45	NA	NA
11	11	23	NA	NA
12	12	23	NA	NA

Number of Products	**m**	9
Number of kits	**n**	7
Reconfiguration Cost	2	
Crossdock Cost	1	

Supply / Demand	
Low	20
High	150

Product Variety	medium
Max Products	20

Bill of Materails	K(1)	K(2)	K(3)	K(4)	K(5)	K(6)	K(7)	DENSITY	NUMBER OF KITS
B(1)	1	2	3	7				**Med dense**	**4**
B(2)	2							**Low dense**	**1**
B(3)	2	7						**Low dense**	**2**
B(4)	1	3	7					**Med dense**	**3**
B(5)	6							**Low dense**	**1**
B(6)	2	5						**Low dense**	**2**
B(7)	1	2	3	4	5	6	7	**High dense**	**7**
B(8)	1	2	3	4	5	6		**High dense**	**6**
B(9)	1	5	6	7				**Med dense**	**4**

On hand inventory of Products	
SP(1)	130
SP(2)	146
SP(3)	75
SP(4)	117
SP(5)	67
SP(6)	38
SP(7)	140
SP(8)	38
SP(9)	123

Demand value for Products	
DP(1)	130
DP(2)	120
DP(3)	99
DP(4)	55
DP(5)	105
DP(6)	114
DP(7)	118
DP(8)	115
DP(9)	26

On hand inventory of Kits	
SK(1)	70
SK(2)	122
SK(3)	64
SK(4)	88
SK(5)	100
SK(6)	25
SK(7)	102

Demand of Kits	
DK(1)	79
DK(2)	48
DK(3)	24
DK(4)	24
DK(5)	112
DK(6)	28
DK(7)	57

On hand inventory of Reclaimed Kits	
SK(1)	66
SK(2)	49
SK(3)	71
SK(4)	117
SK(5)	128
SK(6)	103
SK(7)	38

Figure B.5: Bill of Materials, Inventory and Demand Data for Medium Product Variety Test Instance 2

B.2.2 Test Instance 2

$N_k^R = \{33, 34, 35, 36, 37, 38, 39\},\ N_k^G = \{47, 48, 49, 50, 51, 52, 53\}$

Table B.6: Medium Product Variety Test Instance 2 Network Flow-based Programming Model Output

Donor Product i	Reconfigured Product j	Units	Consuming kits	Releasing	Valid	Trace Origin	Correct Consumption
2	3	12	{39}	∅	Yes	PP1	{39, 53}
2	6	14	{37}	∅	Yes	PP1	{37, 51}
4					No	PP1	
9	3	12	{35, 39}	{33, 37, 38}	No	PP1	{34, 48}
9	8	18	{48, 36}	{39}	No	PP1	{34, 35, 36, 48, 49, 50}
2	3	12	{39}	∅	Yes	PP2	{39, 53}
2	6	14	{37}	∅	Yes	PP2	{37, 51}
9	8	59	{35, 36}	{39}	No	PP2	{34, 35, 36, 48, 49, 50}
1	1	130	NA	NA	NA	NA	NA
2	2	120	NA	NA	NA	NA	NA
3	3	75	NA	NA	NA	NA	NA
4	4	55	NA	NA	NA	NA	NA
5	5	67	NA	NA	NA	NA	NA
6	6	38	NA	NA	NA	NA	NA
7	7	118	NA	NA	NA	NA	NA

Table B.7: Medium Product Variety Test Instance 2 Integer Program based Programming Model Output

Donor Product i	Reconfigured Product j	Units	Consuming kits	Releasing
2	6	26	$\{5\}$	$\emptyset$
4	3	24	$\{2\}$	$\{1,3\}$
4	8	30	$\{2,4,5,6\}$	$\{7\}$
9	6	50	$\{2\}$	$\{1,6,7\}$
9	8	47	$\{2,3,4\}$	$\{7\}$
1	1	130	NA	NA
2	2	120	NA	NA
3	3	75	NA	NA
4	4	55	NA	NA
5	5	67	NA	NA
6	6	38	NA	NA
7	7	118	NA	NA
8	8	38	NA	NA
9	9	26	NA	NA

B.3 High Product Variety

Number of Products	**m**	19
Number of kits	**n**	10
Reconfiguration Cost	2	
Crossdock Cost	1	

Supply / Demand	
Low	20
High	150

Product Variety	high
Max Products	20

Bill of Materails	K(1)	K(2)	K(3)	K(4)	K(5)	K(6)	K(7)	K(8)	K(9)	K(10)	DENSITY	NUMBER OF KITS
B(1)	3	10									**Low dense**	**2**
B(2)	4	6	8								**Low dense**	**3**
B(3)	1	2	3	4	5	6	7	8	10		**High dense**	**9**
B(4)	1	2	3	4	5	6	7	8	9	10	**High dense**	**10**
B(5)	1	6	7								**Low dense**	**3**
B(6)	1	7	9	10							**Med dense**	**4**
B(7)	2	3	4	6	7	8	9	10			**High dense**	**8**
B(8)	2	3	4	5	6	7	8	9	10		**High dense**	**9**
B(9)	6	7	8								**Low dense**	**3**
B(10)	1	2	3	4	5	6	8	9			**High dense**	**8**
B(11)	1	3	4	7							**Low dense**	**4**
B(12)	1	2	3	4	6	7	8	9	10		**High dense**	**9**
B(13)	5	7	10								**Low dense**	**3**
B(14)	1	2									**Low dense**	**2**
B(15)	3	5	9	10							**Med dense**	**4**
B(16)	2	3	4	5	6	7	8	9			**High dense**	**8**
B(17)	1	2	4	6	7	8	9				**Med dense**	**7**
B(18)	1	2	6	9							**Low dense**	**4**
B(19)	3	4	5	6	7	8	9	10			**High dense**	**8**

Figure B.6: Bill of Materials for High Product Variety Test Instance 1

On hand inventory of Kits

SP(1)	140
SP(2)	124
SP(3)	77
SP(4)	60
SP(5)	144
SP(6)	144
SP(7)	70
SP(8)	99
SP(9)	101
SP(10)	146
SP(11)	124
SP(12)	20
SP(13)	126
SP(14)	20
SP(15)	75
SP(16)	23
SP(17)	124
SP(18)	104
SP(19)	124

Demand value for Products

DP(1)	104
DP(2)	83
DP(3)	73
DP(4)	64
DP(5)	52
DP(6)	109
DP(7)	34
DP(8)	130
DP(9)	33
DP(10)	115
DP(11)	50
DP(12)	97
DP(13)	57
DP(14)	22
DP(15)	101
DP(16)	40
DP(17)	57
DP(18)	82
DP(19)	136

On hand inventory of Kits

SK(1)	146
SK(2)	146
SK(3)	27
SK(4)	34
SK(5)	84
SK(6)	55
SK(7)	136
SK(8)	57
SK(9)	68
SK(10)	113

Demand of Kits

DK(1)	147
DK(2)	146
DK(3)	74
DK(4)	53
DK(5)	111
DK(6)	72
DK(7)	86
DK(8)	113
DK(9)	109
DK(10)	49

On hand inventory of Reclaimed Kits

SK(1)	75
SK(2)	113
SK(3)	27
SK(4)	136
SK(5)	126
SK(6)	28
SK(7)	128
SK(8)	130
SK(9)	55
SK(10)	65

Figure B.7: Inventory and Demand Data for High Product Variety Test Instance 1

B.3.1 Test Instance 1

$$N_k^R = \{59, 60, 61, 62, 63, 64, 65, 66, 67, 68\},\ N_k^G = \{79, 80, 81, 82, 83, 84, 85, 86, 87, 88\}$$

Table B.8: High Product Variety Test Instance 1 Network Flow-based Programming Model Output

Donor Product i	Reconfigured Product j	Units	Consuming kits	Releasing	Valid	Trace Origin	Correct Consumption
5	19	12	{66}	{59}	No	PP1	{61, 62, 63, 66, 67, 68, 81, 82, 83, 86, 87, 88, 38}
11	14	2	{60}	{61, 62, 65}	Yes	PP1	{60, 80}
13	12	69	{59}	{63}	No	PP1	{59, 60, 61, 62, 64, 66, 67, 79, 80, 81, 82, 84, 86, 87}
17	8	31	{68}	{59}	No	PP1	{61, 63, 68, 81, 83, 88}
17	15	26	{63}	{59, 60, 62, 64, 65, 66}	No	PP1	{61, 63, 68, 81, 83, 88}
18	16	17	{65}	{59}	No	PP1	{61, 62, 63, 65, 66, 81, 82, 83, 85, 86}
17	4	4	{63}	∅	No	PP2	{88, 83, 81, 6863, 61, 17}
17	8	31	{68}	{59}	No	PP2	{61, 63, 68, 81, 83, 88}
13	12	69	{59}	{63}	No	PP2	{59, 60, 61, 62, 64, 66, 67, 79, 80, 81, 82, 84, 86, 87}
11	12	8	{60}	∅	No	PP2	{88, 87, 86, 84, 80, 68, 67, 66, 64, 60}
11	14	2	{60}	{61, 62, 65}	Yes	PP2	{60, 80}
17	15	26	{63}	{59, 60, 62, 64, 65, 66}	No	PP2	{61, 63, 68, 81, 83, 88}
18	16	17	{65}	{59}	No	PP1	{61, 62, 63, 65, 66, 81, 82, 83, 85, 86}
5	19	12	{66}	{59}	No	PP1	{61, 62, 63, 66, 67, 68, 81, 82, 83, 86, 87, 88, 38}
1	1	104	NA	NA	NA	NA	NA
2	2	83	NA	NA	NA	NA	NA
3	3	73	NA	NA	NA	NA	NA
4	4	60	NA	NA	NA	NA	NA
5	5	64	NA	NA	NA	NA	NA
6	6	109	NA	NA	NA	NA	NA
7	7	34	NA	NA	NA	NA	NA
8	8	99	NA	NA	NA	NA	NA
9	9	33	NA	NA	NA	NA	NA
10	10	115	NA	NA	NA	NA	NA
11	11	60	NA	NA	NA	NA	NA
12	12	20	NA	NA	NA	NA	NA
13	13	126	NA	NA	NA	NA	NA
14	14	20	NA	NA	NA	NA	NA
15	15	75	NA	NA	NA	NA	NA
16	16	23	NA	NA	NA	NA	NA
17	17	118	NA	NA	NA	NA	NA
18	18	99	NA	NA	NA	NA	NA
19	19	124	NA	NA	NA	NA	NA

Table B.9: High Product Variety Test Instance 1 Integer Program based Programming Model Output

Donor Product i	**Reconfigured Product j**	**Units**	**Consuming kits**	**Releasing**
1	15	26	$\{5, 9\}$	$\emptyset$
3	8	4	$\{9\}$	$\{2\}$
7	8	24	$\{5\}$	$\{1\}$
7	19	12	$\{5\}$	$\{2\}$
10	4	4	$\{7, 10\}$	$\emptyset$
10	8	3	$\{7, 10\}$	$\{1\}$
10	12	7	$\{7, 10\}$	$\{5\}$
10	16	17	$\{7\}$	$\{1\}$
11	5	18	$\{6\}$	$\{3, 4\}$
11	12	3	$\{2, 6, 8, 9, 10\}$	$\emptyset$
11	14	22	$\{2\}$	$\{3, 4, 7\}$
17	12	67	{3,10}	$\emptyset$
1	1	104	NA	NA
2	2	83	NA	NA
3	3	73	NA	NA
4	4	60	NA	NA
5	5	34	NA	NA
6	6	109	NA	NA
7	7	34	NA	NA
8	8	99	NA	NA
9	9	33	NA	NA
10	10	115	NA	NA
11	11	50	NA	NA
12	12	20	NA	NA
13	13	57	NA	NA
14	14	0	NA	NA
15	15	75	NA	NA
16	16	23	NA	NA
17	17	57	NA	NA
18	18	82	NA	NA
19	19	124	NA	NA

B.3.2 Test Instance 2

$N_k^R = \{55, 56, 57, 58, 59, 60, 61, 62, 63, 64\}$, $N_k^G = \{65, 66, 67, 68, 69, 70, 71, 72, 73, 74\}$

Table B.10: High Product Variety Test Instance 2 Network Flow-based Programming Model Output

Donor Product i	Reconfigured Product j	Units	Consuming kits	Releasing	Valid	Trace Origin	Correct Consumption
4	12	5	{60}	{55, 58, 59, 63, 64}	Yes	PP1	{60, 80}
4	17	53	57	{55, 56, 59, 62, 63, 64}	Yes	PP1	{57, 77}
7					No	PP1	
7	13	53	{62}	{63, 59}	No	PP1	{62, 64, 82, 84, 30}
9					No	PP1	
11	13	20	{62}	{56, 57, 59, 60, 61, 63}	Yes	PP1	{62, 82}
4	1	2	{60}	{55, 56, 59, 62, 63}	Yes	PP2	{60, 80}
4	2	29	{60}	{64}	No	PP2	{57, 60, 77, 70}
4	3	6	{60}	{61}	Yes	PP2	{60, 80}
4	8	15	{57}	{55, 59, 61, 62, 63}	Yes	PP2	{57, 77}
4	12	5	{60}	{55, 58, 59, 63, 64}	Yes	PP2	{60, 80}
11	13	20	{62}	{56, 57, 59, 60, 61, 63}	Yes	PP2	{62, 82}
7					No	PP2	
9					No	PP2	
4	17	53	{57}	{55, 56, 59, 62, 63, 64}	Yes	PP2	{57, 77}
1	1	64	NA	NA	NA	NA	NA
2	2	40	NA	NA	NA	NA	NA
3	3	123	NA	NA	NA	NA	NA
4	4	24	NA	NA	NA	NA	NA
5	5	104	NA	NA	NA	NA	NA
6	6	61	NA	NA	NA	NA	NA
7	7	67	NA	NA	NA	NA	NA
8	8	50	NA	NA	NA	NA	NA
9	9	54	NA	NA	NA	NA	NA
10	10	109	NA	NA	NA	NA	NA
11	11	98	NA	NA	NA	NA	NA
12	12	33	NA	NA	NA	NA	NA
13	13	58	NA	NA	NA	NA	NA
14	14	83	NA	NA	NA	NA	NA
15	15	106	NA	NA	NA	NA	NA
16	16	37	NA	NA	NA	NA	NA
17	17	35	NA	NA	NA	NA	NA

Number of Products	**m**	17
Number of kits	**n**	10
Reconfiguration Cost	2	
Crossdock Cost	1	

Low	20
High	150

Supply / Demand

Low	20
High	150

Product Variety	high
Max Products	20

Bill of Materails	K(1)	K(2)	K(3)	K(4)	K(5)	K(6)	K(7)	K(8)	K(9)	K(10)	DENSITY	NUMBER OF KITS
B(1)	4	6	7	10							**Low dense**	**4**
B(2)	1	2	3	4	5	6	7	8	9		**High dense**	**9**
B(3)	1	2	4	5	6	8	9	10			**Med dense**	**8**
B(4)	1	2	4	5	7	8	9	10			**Med dense**	**8**
B(5)	5	7	8	10							**Low dense**	**4**
B(6)	1	5	8	9							**Low dense**	**4**
B(7)	4	9	5								**Low dense**	**3**
B(8)	2	3	4	10							**Low dense**	**4**
B(9)	2	3	6	7	8	9	10				**Low dense**	**7**
B(10)	6	7	8	10							**Low dense**	**4**
B(11)	2	3	4	5	6	7	9	10			**High dense**	**8**
B(12)	2	6	7	8							**Low dense**	**4**
B(13)	4	8	10								**Low dense**	**3**
B(14)	2	10									**Low dense**	**2**
B(15)	3	8	9	10							**Low dense**	**4**
B(16)	4	5									**Low dense**	**2**
B(17)	3	4	7								**Low dense**	**3**

Figure B.8: Bill of Materials for High Product Variety Test Instance 2

On hand inventory of Products

SP(1)	64
SP(2)	40
SP(3)	123
SP(4)	134
SP(5)	104
SP(6)	61
SP(7)	133
SP(8)	50
SP(9)	71
SP(10)	109
SP(11)	118
SP(12)	33
SP(13)	58
SP(14)	83
SP(15)	106
SP(16)	37
SP(17)	35

Demand value for Products

DP(1)	66
DP(2)	69
DP(3)	129
DP(4)	24
DP(5)	128
DP(6)	72
DP(7)	67
DP(8)	110
DP(9)	54
DP(10)	109
DP(11)	98
DP(12)	38
DP(13)	150
DP(14)	96
DP(15)	146
DP(16)	91
DP(17)	88

On hand inventory of Kits

SK(1)	111
SK(2)	52
SK(3)	127
SK(4)	85
SK(5)	104
SK(6)	45
SK(7)	75
SK(8)	122
SK(9)	63
SK(10)	34

Demand of Kits

DK(1)	144
DK(2)	37
DK(3)	100
DK(4)	26
DK(5)	73
DK(6)	23
DK(7)	79
DK(8)	57
DK(9)	68
DK(10)	50

On hand inventory of Reclaimed Kits

SK(1)	146
SK(2)	36
SK(3)	123
SK(4)	104
SK(5)	115
SK(6)	116
SK(7)	24
SK(8)	112
SK(9)	37
SK(10)	73

Figure B.9: Inventory and Demand Data for High Product Variety Test Instance 2

Table B.11: High Product Variety Test Instance 1 Integer Program based Programming Model Output

Donor Product i	**Reconfigured Product j**	**Units**	**Consuming kits**	**Releasing**
4	3	6	{6}	{7}
4	8	60	{3}	{1, 5, 7, 8, 9}
4	15	40	{3}	{1, 2, 4, 5, 7}
4	17	4	{3}	{1, 2, 5, 8, 9, 10}
7	13	17	{8, 10}	{9, 5}
7	17	49	{3, 7}	{9, 5}
9	5	9	{5}	{2, 3, 6, 9}
9	13	8	{4}	{2, 3, 6, 7, 9}
11	5	15	{8}	{2, 3, 4, 6, 9}
11	12	5	{8}	{3, 4, 5, 9, 10}
1	1	64	NA	NA
2	2	40	NA	NA
3	3	123	NA	NA
4	4	24	NA	NA
5	5	104	NA	NA
6	6	61	NA	NA
7	7	67	NA	NA
8	8	50	NA	NA
9	9	54	NA	NA
10	10	109	NA	NA
11	11	98	NA	NA
12	12	33	NA	NA
13	13	58	NA	NA
14	14	83	NA	NA
15	15	105	NA	NA
16	16	37	NA	NA
17	17	35	NA	NA

VITA

Jeet Arvind Turakhia

Candidate for the Degree of

Master of Science

Thesis: MODELING PRODUCT RECONFIGURATION AT

DISTRIBUTION CENTER LEVEL

Major Field: Industrial Engineering and Management

Biographical:

Education:

Completed the requirements for the Master of Science in Industrial Engineering and Management at Oklahoma State University, Stillwater, Oklahoma in July, 2013.

Completed the requirements for the Bachelor of Engineering in Mechanical Engineering at Shivaji University, Maharashtra, India in 2010.

Experience:

Grader - School of Industrial Engineering and Management, Oklahoma State University (January 2013 to May 2013).

Graduate Teaching Assistant - College of Engineering, Architecture and Technology, Oklahoma State University (August 2012 to December 2012).

Industrial Engineer (Internship) - Cummins Inc. - Memphis, TN (May 2012 to August 2012).

Graduate Teaching Assistant - Engineering Technology and Management Program, Oklahoma State University (August 2011 to May 2012).

In-Plant Trainee - Menon and Menon Ltd. Kolhapur, Maharashtra, India (December 2008 to January 2009).