12-2012

Integrated Production and Distribution Problem with Single Perishable Product

Wennian Li
Clemson University, li_wennian@hotmail.com

INTEGRATED PRODUCTION AND DISTRIBUTION PROBLEM WITH SINGLE PERISHABLE PRODUCT

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Industrial Engineering

---

by
Wennian Li
December 2012

Accepted by:
Dr. William G. Ferrell, Committee Chair
Dr. Maria E. Mayorga
Dr. Mary E. Kurz
Dr. Byung Rae Cho

ABSTRACT

This dissertation investigated the extension of the Integrated Production and Distribution Scheduling Problem (IPDSPP) using a variety of perishable products, applying the JIP principle and make-to-order strategy to integrate the production and distribution schedules. A single perishable product with a constant lifetime after production was used in the model discussed here. The objective of the problem was to find the solution that results in minimal system total transportation costs while satisfying customer demand within a fixed time horizon. In the solution, the fleet size, vehicle route and distribution schedule, plant production batch size and schedule were determined simultaneously. This research employed non-identical vehicles to fulfill the distribution; each allowed multiple trips within the time horizon. Both the single plant and multiple plant scenarios were analyzed in this research. For each scenario, the complexity analysis, mixed integer programming model, heuristic algorithms and comprehensive empirical study are provided.

DEDICATION

This work is dedicated to my parents and my wife.

ACKNOWLEDGMENTS

I would like to thank the people who helped me to complete the work in this dissertation. I am very grateful to my supervisors William G. Ferrell. I would like to thank Dr. William G. Ferrell for his patience, technical advice, encouragement and insightful comments through my dissertation work.

I would also like to thanks the members of my committee Dr. Maria E. Mayorga, Dr. Mary E. Kurz and Dr. Byung Rae Cho for their constructive criticism concerning the material of this dissertation. Thanks also to all my friends at the IE department and in Clemson for lightening up my life every day.

I would like to express my special thanks to my wife Ling Zu. Her understanding and faith in me, and her love, encouragement, support motivated me all the time. Last but not least, I would like to thank continuous support throughout all my years here at the Clemson University.

TABLE OF CONTENTS

Table of Contents (Continued)

LIST OF TABLES

List of Tables (Continued)

Table Page

List of Tables (Continued)

Table | Page

LIST OF FIGURES

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

List of Figures (Continued)

List of Figures (Continued)

Figure Page

CHAPTER ONE

INTRODUCTION

Given the current trends in globalization, many companies increasingly market their products worldwide. For many of them, success depends on producing items customized to the degree that they satisfy individual requirements but within a short period of time and at a low cost. Over the past two decades, the means to accomplish these contradictory goals has placed an ever-increasing focus on the supply chain. As a result much research both in industry and in academia has focused on the integration of different functions in supply chains to simultaneously increase flexibility, reduce cycle time and keep costs low. This integration not only reduces the number of steps in the process but it also tends to eliminate the inherent obstacles between different functions so that optimization across larger segments of the supply chain can be realized. The research reported here is consistent with this trend, focusing on the integrated production and distribution scheduling problem associated with perishable products (IPDSPPP).

While the initial definition of perishable product may reflect a limited set of items, primarily focusing on food, in reality many products are considered perishable when this property is interpreted in a broader sense. In general, a perishable product is one whose value or quality is closely related to the length of time between when it is ready to be sold and when it is consumed. For example, products like designer clothes and computers that have a relatively long lifetime compared to fresh vegetables, milk and blood. The research reported here considers perishable products those which have full utility and usefulness for a short, fixed period of time after which they are useless. In the context of items that are produced, the item lifetime begins the moment production is complete, meaning they must be consumed before this fixed time span elapsed. It is assumed that the lifetime is much shorter than the planned time horizon.

This short fixed lifetime has several implications. One is that implementing a make-to-order strategy for any phase of a production plan must be strategically planned because the time that a product is held in inventory plus the time spent in transit to the customer must be less than the lifetime. Therefore, creating flexibility and efficiency in this segment of the supply chain requires that the production and distribution schedules be highly integrated. Inefficiency in this integration could result in the product becoming unusable before delivery to the customer is complete or the inability to satisfy the customer demand before the deadline. The former will result in the waste of raw materials production capacity, power, and time and, sometimes a significant cost for safely salvaging the product. The latter leads to a failure to meet the consumer's demand on time, resulting the loss of a customer or, at a minimum, the loss of good will.

This research uses several approaches to conduct a detailed analysis of this integrated production and distribution problem. A two-echelon supply chain is considered with both one supplier and multiple suppliers satisfying demands from multiple customers, both situations being referred to as a supplier in this discussion. A single period is considered, for example, one day. The capacity of the supplier is assumed to be limited and fixed but sufficient to satisfy all customer demands within the specified time horizon. The supplier is also assumed to provide the distribution service for delivering the product to the customer locations. In this research, it is assumed that the supplier determines the production and distribution schedules, renting a fleet of vehicles to implement them rather than using a third party logistics provider.

To create the model, the production control strategy of the supplier is assumed to be make-to-order, meaning that customers place orders in advance. All customers are assumed to act independently and are geographically dispersed on a 2-dimensional space. Furthermore, each customer has a randomly assigned demand that must be satisfied within the time horizon. Early delivery is not penalized, so the suppliers have a larger degree of freedom in this research scenario than they would if customers had precise windows during which the product had to be delivered. These assumptions mean that for a given time horizon (e.g. a given day), the supplier must schedule production to satisfy the deterministic demands of all customers within that time frame and sequence deliveries so that all customers are served within the time horizon with a product whose lifetime has not expired. Since the product has a constant life time after production is completed, the delivery schedule and the production schedule must be integrated to ensure these deadlines are met. These assumptions lead to the geographic restriction that all customers must be located no further than a vehicle can travel during one half of the time horizon because it is assumed that all vehicles must return to the supplier before the time horizon expires.

In addition, the following practical assumptions are also made concerning the distribution: 1) mixed fleet is considered, i.e. all of the rented vehicles are assumed to be non-identical in capacity and cost. 2) each customer's demand is fulfilled with one vehicle during one delivery, 3) vehicles can visit several customers in one trip, 4) each trip begins from the plant, goes through all customer in the trip and then returns to the plant, 5) the vehicle is allowed to carry less than a full load, and 6) after one vehicle finishes a trip, it can be reused to make another one if it can meet the requirements for delivery and return.

The decisions the company must make include determining the number and type of vehicles, their departure time and routing schedule, and the related batch production schedules to minimize the total operating costs while satisfying all customer demands within the time horizon. Since this research considers only a make-to-order strategy, the demand for every period is deterministic, meaning the total cost of production is fixed and only the delivery costs must be considered to minimize the total operating costs. The total distribution cost is composed of two components: 1) the fixed setup cost is the one-time cost paid to use each vehicle for the distribution of the deliveries over the fixed time horizon. For example, if the time horizon is one day, the fixed setup cost is the sum of the daily fee to rent each vehicle, plus such costs as those for a driver that are incurred for the entire day regardless of the number of trips made or length of each trip. 2) The variable operating cost is dependent on the distance traveled which includes the cost of fuel plus the vehicle maintenance fee for the distance traveled. It is calculated by multiplying the variable cost of vehicle by the total distance traveled. Since this research uses non-identical vehicles for distribution, the calculation of the distribution cost is even more complex. Each type of vehicle has a unique capacity, fixed setup cost and variable cost. For example, different types of vehicles incur different rental fees, driver salaries, and fuel efficiencies depending on size/capacity and fuel type, meaning different vehicle types have different variable costs for the same distance traveled. In addition to the system transportation cost, the feasible solution must include the routing schedule for each vehicle that meets all constraints such as vehicle capacity, product lifetime, and delivery before the end of the time horizon. At the same time, the batch production schedule in the solution should be consistent with the related delivery schedule.

The problem addressed in this research is an extension of the classical Vehicle Routing Problem (VRP) which has been shown to fall within the class of nondeterministic polynomial-time hard problems (NP-hard problem). This means that all known algorithms that define an optimal solution require exponentially increasing computational time as the number of customers increase; therefore, heuristic methods which provide approximate solutions are justified and are required for realistic sized problems. As such, mixed integer programming (MIP) is used for smaller sized problems and heuristics are employed to solve larger and more realistic sized problems.

CHAPTER TWO

LITERATURE REVIEW

The integration of various functions in the supply chain is currently the focus of much supply chain management research, especially since it has proven profitable for such companies as Dell and Wal-Mart. Specifically, integrating production and distribution, two important functions in the supply chain, can be beneficial. Some of the research on this integration focuses on a global perspective, providing such valuable reviews as those by Vidal and Goetschalckx (1997), Sarmiento and Nagi (1999), Erenguc et al. (1999) and Chen (2004). Vidal and Goetschalckx (1997) review the literature on the strategic production-distribution models, classifying them as domestic and international and focusing on the analysis of several mixed integer programming models to determine the relevant factors in the formulations. Owen and Daskin (1998) consider the transportation system in their review of the literature, focusing on the logistics and benefits of integration, while Erenguc et al. (1999) divide the integration of production and distribution into 3 stages, supply, plant and distribution, concluding that the decisions at each needs to be made jointly to optimize the network. Chen (2004) provides a detailed classification of the problem in three different dimensions, the decision level (tactical and operational level), the integration structure (inbound and outbound transportation), the problem parameters (single period, finite period and infinite period). The research reported focuses on finding an efficient algorithm for the integrated production and distribution problem with a perishable product. Thus, the relevant literature is divided into 2 categories: (1) the traditional integrated production and distribution problem, and (2) the integrated production and distribution problem with a perishable product. The detailed discussion of these two areas is based on the decision level (tactical and operational level) for each function (production and distribution).

Regular Integrated Production and Distribution Problem

Much research has been conducted on the integrated production and distribution problem without considering the lifetime of the product. The integrated model reveals different levels of details for each of these functions depending on the level of decision making (strategic, tactical, or operational).

For the strategic level integration model, the integrated production and distribution problem is considered under the explicit system, with the model analyzing only the strategic or tactical level decisions for both functions. This model usually integrates the production and distribution using a multiple plant and multiple customer scenario. The production cost for each plant, which is independent of the others, is usually represented by a concave function which is dependent on the amount of production at the plant. The distribution commonly uses a direct delivery strategy to transport the product from the plant to the customer, with the relative costs for one unit of product transported between the i^{th} plant and j^{th} customer being fixed. The total distribution cost changes based on the quantities assigned for the N plants and the M customers. The objective of this type of problem is to coordinate the production quantity in each plant and distribute the quantity between each plant and its customers to minimize the total system cost, which is the sum of the total production and distribution costs. Palekar et al. (1990) investigated a branch and bound method to solve this integrated production and distribution problem with multiple plants and multiple customers. In this method, each plant has a limited capacity and the direct delivery and production costs between the i^{th} customer and the j^{th} plant are fixed. The system objective is to find the strategy that serves all customers while minimizing the total cost. Tuy et al. (1993) consider a similar problem using a 2 plant, M customer scenario. Unlike the previous research, the production cost is a concave function related to the amount of production at each plant. A polynomial time algorithm was proposed to solve the problem. In more recent research, Tuy et al. (1996) extended the previous model from two plants to consider a multiple plant scenario.

Youssef and Mahmoud (1996) examined solution strategies for the production-distribution problem by comparing several heuristic approximation algorithms, concluding that the trade-off between the transportation and production costs will lead the solution toward centralization. Kuno and Utsunomiya (2000) proposed a Lagrangian relaxation to generate improved bounds to accelerate the search for a solution using the branch and bound algorithm. More details about this basic integrated model can be found in Vidal and Goetschalckx (1997) and Chen (2004).

More recently, the research being conducted has included detailed production and distribution systems, some of which extends the production scheduling aspect by considering the classical Schedule Problem (SP) and the transportation component by considering an explicit system such as immediate direct delivery, fixed transportation times or linear transportation costs. Potts (1980) provided a heuristic algorithm for a single machine scheduling problem with a fixed transpiration time, the objective being to minimize the total makespan (i.e. the latest time for all customer demand to be satisfied). Immediate direct delivery, infinite vehicle capacity and fixed transportation time were applied in this study. Similar models have been developed by various researchers using an explicit transportation system, for example, Hall and Shmoys (1992), Woeginger (1994, 1998), and Zdrzałka, (1995). Strusevich (1999) developed a heuristic algorithm for the two-machine, open-shop problem for minimizing the total makespan. The product transportation time between different machines is considered in this model, and the transportation decision is integrated into the production scheduling. Lee and Chen (2001) considered the transportation time both for production, which includes the time for the product to move between machines, and the distribution, which involves the delivery of the final product to the customer. A constraint limiting the capacity of the vehicles was applied in the model. Chang and Lee (2004) extend this research using three special cases: single machine, single vehicle and single customer; parallel machines, single vehicle and single customer; and single machine, single vehicle and two customers. They provided the heuristic algorithms and analysis of the worst case for each scenario. Li et al (2004, 2005, 2006) and Zandieh and Molla (2009) investigated the issues involving synchronizing production and air transportation. Considering the capacity and time penalties, they modified the objective to minimize the total transportation cost rather than minimize the system makespan, but the transportation system remained explicit with direct delivery.

Another focus of much research considers detailed scheduling of the distribution combined with the classical transportation Vehicle Routing Problem (VRP). In this scenario, the production is usually simplified by assuming either a fixed production rate for each period or instantaneous production. The production cost is commonly represented by the cost of the product and is often connected with the inventory and the out stock penalty costs. Chandra and Fisher (1994) considered a network with a single production center, multiple retailers, and several products. A fleet of vehicles was used to deliver the product to multiple retailers. For each period, the solution needed to determine the production quantity for each type of product and to schedule the vehicle routes for the entire fleet to meet all delivery demands. Multiple periods were considered in this research. The objective was to schedule the production and distribution to minimize the total cost of the production setup, transportation and inventory over multiple periods. Fumero and Vercellis (1999) extended this model by introducing limited production and distribution resources. They solved the problem using the Lagrangean relaxation method, comparing the integrated decision (a synchronized decision for the production and distribution schedule) and the separate decision (a sequential decision for the production and distribution schedule). Park (2005) extended this model to a multiple plant scenario, with a heuristic algorithm being provided for this integration problem with multiple plants, multiple items, multiple retailers and multiple periods. In addition, a sensitivity analysis was used to analyze the input variables.

Detailed consideration of both the production and distribution result in the most complex combination of the integrated production and distribution problem. Given the complexity of this problem, only a limited amount of research has been conducted in this area, most of it in the last few years. Garcia et al. (2004) investigated the integrated production and distribution problem using a multiple plant situation. Each plant had a fixed capacity and the product was delivered immediately once production was finished. The customers were separated geographically with independent orders associated with different profits and deadlines. A fixed number of vehicles were used to deliver the products among multiple customers and plants. The objective was to maximize the profit of the solution schedule considering both the profitable value of the served orders, the overdue penalty cost and the transportation costs. Chen and Vairaktarakis (2005) researched this integrated problem, applying it to the computer and food catering industries. A set of jobs were first produced in the production center and then delivered to the customer directly without holding intermediate inventory. The customer service level was measured by the amount of time needed to deliver the product to the customer. The purpose of the research was to find the solution schedule that optimized the objective function, which was the combined customer service level and transportation costs. Heuristic algorithms were developed and tested using several combination scenarios (i.e. single machine scenario and parallel machine scenario in production scheduling component and the maximum time or average time measurement for customer service level).

Integrated Production and Distribution Problem with Perishable Product

Problems in this class are usually concerned with perishable products having a short lifetime. Most research in this area considers the loss cost of the perishable product in the inventory component, integrating the inventory decision into the system. The research reported here focuses on the integrated production and distribution problem without intermediate inventory consideration. The short lifetime requires that the production and distribution schedules be highly integrated, which, in turn, requires detailed consideration of each. A recent review by Chen (2009) describes this category as the most complex of the integrated production and distribution problems to model. Several recent researches on this problem had been conducted by Armstrong et al. (2008), Geismar et al. (2008) and Devapriya et al. (2008).

Armstrong et al. (2008) considered a perishable item production and distribution problem with a single machine production and a single delivery vehicle serving multiple customers. All orders were delivered in a preplanned customer sequence, and the constraints included a constant lifespan for the perishable product and customer requirements regarding quantity and time windows within which the delivery was to be received. Since in this model all demands cannot be satisfied, the objective was to find the subset of the customers which could be served to maximize the demand that could be met. The authors showed that this problem is NP-hard, and a branch-and-bound search algorithm as well as a heuristic lower bound was developed.

Similarly, Geismar et al. (2008) considered a system that includes a single machine and a single vehicle to produce and deliver the perishable product. The plant, which must satisfy all customer demands, has the flexibility to determine the sequence of the production and distribution of each order. The customer does not have a time window, but the perishable product has a constant lifetime meaning every order must be delivered to the customer before it expires. The objective was to find the minimum time span that satisfied all customer orders. A two-phase heuristic method was proposed for this strong NP-hard problem.

Devapriya et al. (2008) extended the work of Geismar et al. (2008) by considering a fixed time horizon and a limited fleet size of identical vehicles that can make multiple trips in a single planning horizon for distribution. As before, all orders must be satisfied within the time horizon. The objective was to determine the minimum fleet size, production and distribution schedules to minimize the total system distribution cost. The authors proposed two heuristics to solve this strong NP-hard problem.

The research reported here extends Devapriya's model in several aspects: first, it introduced non-identical vehicles into the model; with different types of vehicles (different capacity, fuel cost rate, rent fee), the model not only can find more optimal solutions but it is also closer to a real-world scenario. Second, the lower bound for the model was improved by considering the inner travelling distance in the vehicle travel trip. This improvement is significant as the number of customers increases. Third, the two-plant scenario was extended to a multiple plant scenario, which was found to be an NP problem even if the single plant scenario can be solved within the polynomial time. Fourth, several heuristic algorithms were developed to solve the problem and improve the efficiency and quality of the previous algorithm. Fifth, the model was analyzed using scenario partition data. In addition, several suggestions for different scenarios are provided.

CHAPTER THREE

SINGLE PLANT INTEGRATED PRODUCTION AND DISTRIBUTION SCHEDULE PROBLEM WITH A FIXED LIFETIME PERISHABLE PRODUCT

Abstract

The integrated production and distribution scheduling problem (IPDSP) is the generic name that has been attached to a class of make-to-order production problems in which the objective is to optimally coordinate the production scheduling and transportation routing problem so that costs are minimized. This research adds the unique feature that the product is perishable. Specifically, products are assumed to be produced in batches and from the moment a batch is completed, there is a finite amount of time during which the product can be delivered to the customer and retain its usefulness. The objective is to determine the production scheduling and transportation routing (number of vehicles and routes for each) to satisfy a set of known customer demands distributed over a geographic region to minimize the total transportation cost. An integer programming model has been developed that includes the following practical assumptions: 1) there is a time horizon within which all demands must be satisfied, 2) several types of delivery vehicles are available to be used each with a unique capacity and cost, 3) vehicles may make multiple stops on a single trip and may make multiple trips within the time horizon. Small numerical examples with 7 customers or fewer are included that can be solved exactly with GUROBI, a commercially available optimization software. A lower bound is provided that considers the inner travelling distance between

customers and idle vehicle waiting time at the beginning of the time horizon. Finally, several heuristic algorithms are offered for larger problems (i.e. more than 7 customers) and their performance is compared to the optimal solution when possible and to each other for larger problems.

Introduction

Production, inventory, and distribution are three important topics in the design and control of a supply chain. Each of these is highly related to the others but, frequently, they are treated independently which can lead to high inefficiency in terms of excess inventory, long cycle times, and high total system cost. Integrating these decisions in the supply chain could simultaneously reduce the total cost of the supply chain; shrink the cycle time, and increase flexibility and efficiency. The Integrated Production and Distribution Schedule Problem (IPDSP) not only integrates these two functions in the supply chain, it also implements a strategy of holding minimum inventory to simplify inventory decisions and makes the internal inventory operation of negligible cost when compared to the other items in the supply chain.

This research focuses on the IPDSP problem but with the unique element that the product is perishable with a finite lifetime which we henceforth will refer to as IPDSPPP. This connection is motivated by the fact that a key feature of IPDSP is that the solution tends to reduce the total cycle time from production at a plant to delivery at the customer's site and short cycle times have a significant impact on industries associated with perishable or time sensitive products. The value of the perishable product is generally modeled in one of two ways: 1) a characteristic of the product is acceptable for a period of time during which is retains full value but after which it is unacceptable and must be scrapped, and 2) the characteristic's quality degrades with time as does its value. Regardless, timely production and distribution is important in all industries but particularly so when perishable products are involved.

Ghare and Schrader (1963) first introduced the perishable idea into the decay of inventory items. Since then, many papers have been published that include perishable products into the supply chain research area. According to Nahmias (1982), perishable products can be classified into two categories: fixed lifetime and random lifetime. Fixed lifetime products have a constant, finite interval of time after production during which time they can be used to satisfy customer demand. The lifetime is assumed to expire after this fixed time regardless of whether the product is held in inventory or in transit to the customer. The idea is that a product with age exceeding the lifetime possesses no value and cannot be used to satisfy customer demand. Random lifetime products perform exactly like fixed lifetime products except the lifetimes follow a probability distribution such as an exponential or Erlang, and the decays continuously and proportionally with time. In this case, the product expires and cannot be used to satisfy customer demand at different ages.

IPDSPPP focuses on products with a short, fixed lifetime. Since the lifetime is limited, the production schedule and distribution plan must be highly integrated to ensure that all customer demand can be satisfied within the time horizon and all products can be delivered to the customer before it expires. Some decisions must be made to minimize the total operation cost for the system such as the total number and type of the delivery vehicles to be used, the departure time of each delivery trip, the routing of each vehicle for every trip, and the plant production schedule and sequence. At the same time, the following practical assumptions are made to more closely model a real system: 1) there is a time horizon within which all demands must be satisfied, 2) there are multiple different types of vehicles used for delivery and each type has a unique capacity and cost structure, 3) vehicles are allowed to make multiple stops on a single trip and make multiple trips within the time horizon.

This research provides several significant features:

- The non-identical vehicle assumption is considered in this research. Mixed fleet includes multiple types of vehicles (different capacities as well as different fixed costs and average cost per mile) are considered.
- A new mathematical programming model is proposed that captures the IPDSPPP when vehicles with different capacities and costs can be used.
- The lower bound of the solution has been improved by considering the inner travelling distance between customers and the vehicle idle time at the beginning of the time horizon.
- A heuristic structure had been developed to solve the IPDSPPP problem.
- Five heuristics algorithms have been developed to find approximate solutions to this problem. The performances of the heuristic algorithms have been compared.

This chapter is organized as follow: section 2 is a brief literature review of the most relevant papers, section 3 provides details of the model, section 4 presents a few numerical examples, and section 5 offers conclusions and recommendations for future work.

Literature Review

From the introduction it is clear that this research has a natural relationship with the vehicle routing problem (VRP) and its extensions. The VRP problem was first introduce by Dantzig, Fulkerson, and Johnson (1954) and has since received much attention. For example, Bodin (1975), Bodin and Golden (1981), Ronen (1988), Min, Jayaraman, and Srivastava (1998), Toth and Vigo (2002), Eksioglu, Vural and Reisman (2009) are reviews of papers in the VRP area. The first work to address multiple trips within the vehicle routing problem framework was Salhi (1987); this feature added significant realism and practicality to the basic problem. Later, Taillard et al. (1996) provide a tabu search algorithm to solve the vehicle routing problem with multiple trips using a bin packing algorithm to assign the routes into vehicles. Brandao and Mercer (1998) consider a mixed fleet for the problem, a three phase heuristic (insertion, tabu search improvement, and restoring feasibility) to solve the problem. Petch and Salhi (2003) considered the problem with a fixed number of vehicles and an objective of minimizing the maximum tardiness for all vehicles. The vehicle routes are generated using a cost saving-based heuristic followed by local search to improve the solution. Jeon et al. (2007) present an adaptive, memory-based heuristic where “memory” is comprised of multiple route solutions.

In summary, this literature represents a wide range of models with differing parameters and scenarios as well as approaches to finding a solution. It is important to note that all of these papers focus exclusively on transportation operations and do not integrate it with production scheduling. As such, they provide valuable support for this work but the actual overlap in the research problem is minimal.

The problem more closely related to this research is the IPDSP which has been the focus of some previous research. Vidal and Goetschalckx (1997), Sarmiento and Nagi (1999), Bilgen and Ozkarahan (2004) and Chen (2004, 2009) provide reviews of the research in this area. Since the problem is complex, earlier research tended to focus on the high level decisions for production and distribution as pointed out in the earlier review papers by Vidal and Goetschalckx (1997) and Sarmiento and Nagi (1999). Later research extended these ideas to a more detailed level for the production and distribution functions. It is noteworthy that most of the previous work does not focus on operational level decisions. Some consider only strategic or tactical level decisions for production and distribution while others have one part operational and the other tactical. Chen (2004, 2009) points out that only a few papers consider the IPDSP with both parts at the operational level; the number is even smaller when considering a perishable product. The most recent papers on IPDSPPP are contributed by Geismar et al. (2008), Armstrong et al (2008), and Devapriya (2008).

Geismar et al. (2008) considered the problem with a single machine for production and a single capacity vehicle to deliver a single type of perishable product. The plant is assumed to have a fixed capacity and there is an unlimited time horizon to satisfy all customer demands. Customers do not have time windows for delivery but the perishable product has a fixed lifetime which restricts the distance that a vehicle can travel because the product must be delivered to customers before its lifetime expires. Beyond the travel time limitation for the delivery trip, the total quantity for one trip cannot exceed the vehicle capacity. The objective is to find the minimum time span to satisfy all customers' demand. A two-phase heuristic method and lower bound are proposed for this strongly NP-hard problem.

Armstrong et al. (2008) consider a similar scenario with a single machine, single delivery vehicle, multiple customers and a perishable product. The difference is that they assume that all orders are delivered in a preplanned sequence and the total customer demand is beyond the plant's capacity. Also, the product, with a constant lifetime, needs to be delivered to the customer before it expires within their time windows. The objective is to find the subset of the customers which can be served to maximize the demand that can be met. The authors show that this problem is NP-hard and a branch-and-bound search algorithm as well as a heuristic lower bound is developed.

Devapriya (2008) extended this research with a single machine, multiple vehicles, multiple customers and a fixed time horizon. He assumed that the vehicles used for distribution can make multiple trips as long as all orders are satisfied within the planning horizon. The objective is to determine the minimum fleet size, their route and the production schedule to minimize the total system distribution cost. Two heuristics are proposed to solve this strongly NP-hard problem.

Mixed Integer Programming Model for the IPDSPPP

Mathematical Model

The problem addressed in this research is a single-period, finite planning horizon scenario with a single production plant ***P*** that produces products which are delivered to n customers in a multi-customer, supply-dispatching network; hence, this problem is a 1-to-n network. The following notation is used in the model:

Indices:

N'	$= \{1,\ldots,n\}$	the set of n customers
N	$= \{0,1,\ldots,n\}$	the set of n customers plus the single plant identified as 0
S	$= \{1,\ldots,s\}$	the set of s different types of vehicles
T	$= \{1,2,\ldots,H\}$	time periods up to the planning horizon, H

Parameters:

r	= fixed plant production rate (units/unit time)
H	= planning time horizon (unit time)
B	= constant lifetime of the final product (unit time)
q_i	= the demand of customer i, where $i \in N'$ (units)
C_s	= capacity of the s^{th} type of vehicle, $s \in S$ (units)
F_i	= fixed dispatch cost per s^{th} type of vehicle, $s \in S$ (\$/vehicle)
R_i	= traveling cost per unit distance with s^{th} type of vehicle, $s \in S$ (\$/unit distance)
$\tau_{i,j}$	= traveling time between customer/plant i and customer/plant j, $i, j \in N'$ (time)

The plant produces the final product for customers and is responsible for delivering the product to each customer's location. Production occurs in batches and the product is perishable which, in this case, means that the final product has a fixed lifetime that begins immediately after production of a batch is completed. The product can only be used to satisfy customer demand before expiration, and there is no salvage value for the expired product. Obviously, this implies that delivery of the product must be completed before the lifetime expires. It is further assumed that all customer demand must be satisfied before the end of the planning horizon and that the plant has a fixed production rate that is sufficiently large to satisfy all customers demand within the planning horizon.

A fleet of non-identical vehicles is used to distribute product to customer locations and each vehicle must finish all delivery and return to the original plant within the time horizon. A vehicle's trip begins at the plant ***P*** (location 0), visits every customer on the trip, and then goes back to the plant ***P*** at the end of the trip. Each customer's demand is assumed deterministic and known. Split delivery is not allowed; that is, demand at each customer location must be accomplished by one vehicle during one stop. This implies that the demands at each customer are restricted to the capacity of the vehicle with the largest capacity, and it is also assumed that at least one customer has a demand small enough for the smallest capacity vehicle to satisfy it, i.e. $\min_{s \in S} C_s \geq \min_{i \in N} q_i$ and $\max_{s \in S} C_s \geq \max_{i \in N} q_i$.

Several additional assumptions are made related to the vehicles: 1) the time required to load and unload the product is not considered. 2) All vehicles travel at a constant speed which is 1 unit distance/unit time. Note that this assumption simplifies the computation examples but does not limit the generalizability of the model. 3) Vehicles can make multiple trips as long as all constraints can be met such as the delivery before product lifetime expires and the trip can be completed and the vehicle returned to the plant before the end of the planning horizon.

The objective in this research is to minimize the total transportation cost while ensuring that all customers' demands are satisfied. Intuitively, this will require simultaneously making decisions on the type and number of vehicles used to deliver the product and relative vehicles' travel routes. At the same time, there are constraints on vehicle capacity, product lifetime, and the deadline of all deliveries accomplished within the time horizon. Hence, the model must simultaneously consider the production schedule, fleet size, type of vehicles in the fleet, route of each vehicle, and the sequence of the deliveries.

The ability for all vehicles to make multiple trips within the time horizon adds complexity to the problem. Since the customer locations and demands are known *a priori*, the travel time from ***P*** to all possible first customers, $\tau_{0,i}$, $i \in N'$, and the travel time from customer/plant *i* to customer/plant *j*, $\tau_{i,j}$, $i, j \in N$, $N = \{0,1,2,...,n\}$ are known. A practical implication is that all customers' demands must be met so a constraint is that it be possible for a truck to travel to each customer before the product lifetime expires, or $\tau_{0,i} < B$, $\forall i \in N'$. Also, all customer demand must be produced within the time horizon, or $\sum_{i \in N'} q_i / r < H$. Finally, after a vehicle delivers to the last customer in the route, it is required to return to the plant before the end of the time horizon so there is another limitation related to the planning horizon, namely, $2\min_{i\in N'}\tau_{0,i}+\sum_{i\in N'}q_i/r<H$.

It is assumed without loss of generality that all vehicles keep a constant speed equal to 1 unit distance/unit time. With this assumption, different types of vehicles will use the same time but different variable cost to finish the same delivery trip.

To formulate a mathematical model the following definitions are made:

Decision variables:

$X_{s\,i\,j\,k\,m}$ = 1 if m^{th} vehicle of type s visits the customer j immediately after customer i on its k^{th} trip, 0 otherwise. $i,j\in N, and\ k,m\in N'$

$P_{k\,m\,t}$ = 1 if the plant is producing for the k^{th} trip of m^{th} vehicle at time epoch t, 0 otherwise. $t\in T, and\ k,m\in N'$

$y_{s\,m}$ = 1 if the m^{th} vehicle in type s is used for delivery, 0 otherwise. $m\in N', s\in S$

$Z_{s\,k\,m}$ = 1 if the m^{th} vehicle of type s is used for k^{th} trip, 0 otherwise. $k,m\in N'$

$t^{d}_{s\,m\,t}$ = Distribution start time of the k^{th} trip of the m^{th} vehicle of type s, $k,m\in N'$

$t^{d}_{s\,m\,t}$ = Production start time of the k^{th} trip of the m^{th} vehicle of type s, $k,m\in N'$

e_i = Arbitrary real numbers to eliminate sub-tours. $i\in N'$

With these variables, the following model is proposed for the IPDSPPP problem:

$$Min \quad \sum_{s\in S}\sum_{i\in N}\sum_{j\in N}\sum_{k\in N'}\sum_{m\in N'}\left(X_{s\,i\,j\,k\,m}\cdot\tau_{i\,j}\cdot R_s\right)+\sum_{m\in N'}\sum_{s\in S}\left(F_s\cdot y_{sm}\right)$$

Subject to:

$$\sum_{i\in N}\sum_{j\in N}X_{s\,i\,j\,k\,m}\cdot q_j\le C_s \qquad \forall\, k,m\in N',s\in S \tag{1}$$

$$\sum_{i\in N}\sum_{k\in N'}\sum_{m\in N'}X_{s\,i\,j\,k\,m}=1 \qquad \forall\, j\in N',s\in S \tag{2}$$

$$\sum_{j\in N}\sum_{k\in N'}\sum_{m\in N'}X_{s\,i\,j\,k\,m}=1 \qquad \forall\, i\in N',s\in S \tag{3}$$

$$\sum_{j\in N'}\sum_{k\in N'}X_{s\,0\,j\,k\,m}=\sum_{k\in N'}Z_{s\,k\,m} \qquad \forall\, m\in N',s\in S \tag{4}$$

$$\sum_{i\in N'}\sum_{k\in N'}X_{s\,i\,0\,k\,m}=\sum_{k\in N'}Z_{s\,k\,m} \qquad \forall\, m\in N',s\in S \tag{5}$$

$$\sum_{i\in N}X_{s\,i\,j\,k\,m}=\sum_{i\in N}X_{s\,jikm} \qquad \forall\, m,k,j\in N',s\in S \tag{6}$$

$$e_i-e_j+1\le n\cdot\left(1-X_{s\,i\,j\,k\,m}\right) \quad \forall i,\,j,k,m\in N',s\in S \tag{7}$$

$$\sum_{s\in S}y_{ms}\le 1 \qquad \forall\, m\in N' \tag{8}$$

$$X_{s\,i\,j\,k\,m}\le y_{s\,m} \qquad \forall\, i,j\in N,\quad k,m\in N',s\in S \tag{9}$$

$$X_{s\,i\,j\,k\,m}\le Z_{s\,k\,m} \qquad \forall\, i,j\in N,\quad k,m\in N',s\in S \tag{10}$$

$$t^{d}_{s\,k\,m} - \left(t^{p}_{s\,k\,m} + \frac{1}{r} \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot q_j \right) + \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot \tau_{i\,j} \leq B \quad \forall\, k, m \in N', s \in S \tag{11}$$

$$t^{d}_{s\,k\,m} + \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot \tau_{i\,j} \leq H \qquad \forall\, k, m \in N', s \in S \tag{12}$$

$$t^{p}_{s\,k\,m} + \frac{1}{r} \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot q_j \leq t^{d}_{s\,k\,m} \qquad \forall\, k, m \in N', s \in S \tag{13}$$

$$t^{d}_{s\,k\,m} + \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot \tau_{i\,j} \leq t^{d}_{s\,k+1\,m} \qquad \forall\, k, m \in N' \setminus \{n\}, s \in S \tag{14}$$

$$t^{p}_{s\,k\,m} + \frac{1}{r} \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot q_j \leq t^{p}_{s\,k+1\,m} \qquad \forall\, k, m \in N' \setminus \{n\}, s \in S \tag{15}$$

$$\sum_{s \in S} \sum_{k \in N'} \sum_{m \in N'} P_{s\,k\,m\,t} \leq 1 \qquad \forall\, t \in T \tag{16}$$

$$\sum_{t \in T} P_{s\,k\,m\,t} \geq \frac{1}{r} \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot q_j \qquad \forall\, t \in T, s \in S \tag{17}$$

$$t^{p}_{s\,k\,m} \leq t \cdot P_{s\,k\,m\,t} + M \cdot \left(1 - P_{s\,k\,m\,t}\right) \qquad \forall\, k, m \in N', \quad t \in T, s \in S \tag{18}$$

$$t^{p}_{s\,k\,m} + \frac{1}{r} \sum_{i \in N} \sum_{j \in N'} X_{s\,i\,j\,k\,m} \cdot q_j \geq t \cdot P_{s\,k\,m\,t} \qquad \forall\, k, m \in N', \quad t \in T, s \in S \tag{19}$$

$$X_{siikm} = 0 \qquad \forall\, k, m \in N', \quad i \in N \tag{20}$$

Constraint (1) is the vehicle capacity constraint for each trip. Constraints (2) and (3) ensure the each customer will be visited once and only once. Constraints (4) and (5) force each vehicle to have the same number of entry plant time and leave plant time, which is also equal to the complete trips time (i.e., leave plant → deliver product to one or more customers → return to plant) as the vehicle had been assigned to. Constraint (6) is the connection constraint for each trip, it ensures that there is no break inside of each trip. Constraint (7) is the sub-tour elimination constraint. Constraints (8), (9) and (10) restrict each vehicle to be of only one type and enforce this on the route-defining variables. Constraint (11) ensures that a delivery is finished before the product expires and (12) forces every delivery to be completed before the time horizon ends. Constraints (13) and (14) ensure that a delivery cannot begin before the product is ready and that the delivery truck cannot start a new trip until it has returned to the plant from its previous trip. Constraint (15) forces production in the sequence of delivery and forbids preemption. This constraint forbids other high priority batch stop current production batch with low priority and produces the new batch with higher priority instead. Constraint (16) restricts the plant to only produce one batch at any given time period. Constraint (17) ensures the plant productivity level will not exceed it maximal productivity level. Constraints (18) and (19) together ensure the plant continues to produce one batch until the entire quantity required for a trip is completed. Constraint (20) prevents a vehicle to from cycling which means traveling does not begin and end at the same customer.

Numerical Examples

Even a cursory review of the model indicates that the number of variables and constraints will increase quickly as the number of customers increase. To explore this a bit further with the goal of solving smaller problems to optimality, consider the decision variables, X_{siikm}, that are used to determine the route of each vehicle for each trip. The total number of route decision variables will increased according to the customer number, n, with the rate of n^4. For the constraints related with the route decision variable, such as $\sum_{i \in N} \sum_{j \in N} X_{sijkm} \cdot q_j \leq C_s$, the total number of constraints will increase with the rate of n^2. The IPDSPPP problem is an extension of the VRP problem; hence, the IPDSPPP problem is an NP-hard problem. Specifically, consider the problem when the product lifetime equals the time horizon and the plant production capacity is very large (i.e. compared to the transportation time, the production time is negligible). In this case, the optimal solution for the IPDSPPP problem is the solution for the VRP problem with multiple trips. The added features of the IPDSPPP problem in this research do not impact this core observation so the model here is also NP-hard.

To perform computational studies, it is possible to solve problems with a small number of customers to optimality using software, such as GUROBI and CPLEX. An experimental study was conducted to determine the maximum number of customers that could be included and have the model find an optimal solution with for a given hardware and software configuration. Examples were generated for 3, 4, 5, 6, 7, 8, 9 10, 15 and 20 customers that consist of randomly generated customer demands and randomly generated locations on the 2-dimensional surface representing the feasible delivery area around the plant. This randomly generated data is checked to ensure all the basic assumptions are satisfied and subsequently modified if they are not. For example, the demand for each customer is restricted to be smaller than the maximum capacity of the largest truck so that all demands can be satisfied in a single trip. Also, the total customer demand is less than the total capacity of the plant during a time period that allows deliveries to be made. There is one additional adjustment to the data that adds to the richness of the problem, namely, at least one customer has demand that can be satisfied by the smallest capacity vehicle.

AMPL and Gurobi 3.0 were used to solve the integer program with a Dell workstation computer (Dell Optiplex 755; 2.33G Hz Intel Core 4 Duo CPU with 4096 gigabytes dual channels memory). When the software is allowed to run until stopped by the default settings of the MIP gap being less than 0.0001 or some other interrupt such as out of memory, the ten examples took an average of 7 hours to find a solution or terminate. The MIP gap is computed as:

$$MIP\ gap = \frac{Current\ Best\ Integer\ Solution - MIP\ Lower\ Bound\ Given\ by\ Gurobi}{Current\ Best\ Integer\ Solution}$$

These results are presented in Table 3.1. The ten examples were rerun using the same termination conditions and the additional condition of no more than an elapsed time of 1 hour. The data from these runs are also shown in Table 3.1 and can be compared with the first set. Two figures are generated from this data - Figure 3.1 shows the total number of variables and constraints in the datasets and Figure 3.2 shows the MIP gap when forced to terminate in 1 hour (elapsed time) and when allowed to terminate on the internal default settings or interrupt (e.g. out of memory). Notice that the optimal solution can be found for problems with up to 7 customers; however, beyond this the optimization software terminates because of out of memory.

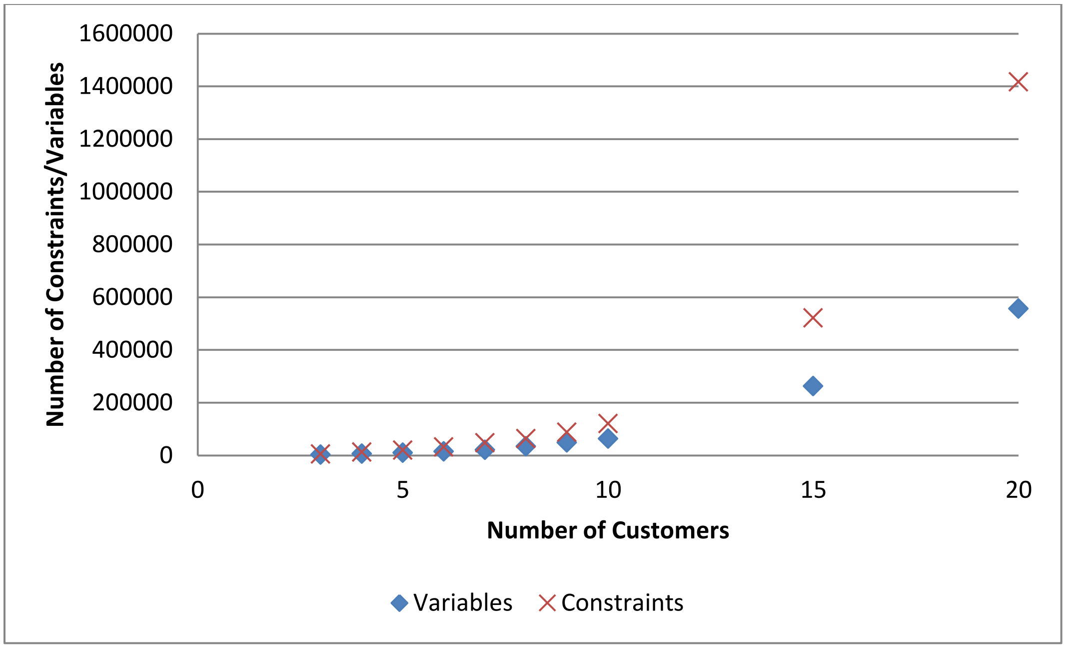

Figure 3.1: Number of Constraints/Variables of MIP Model of IPDSPPP for 10 Problems

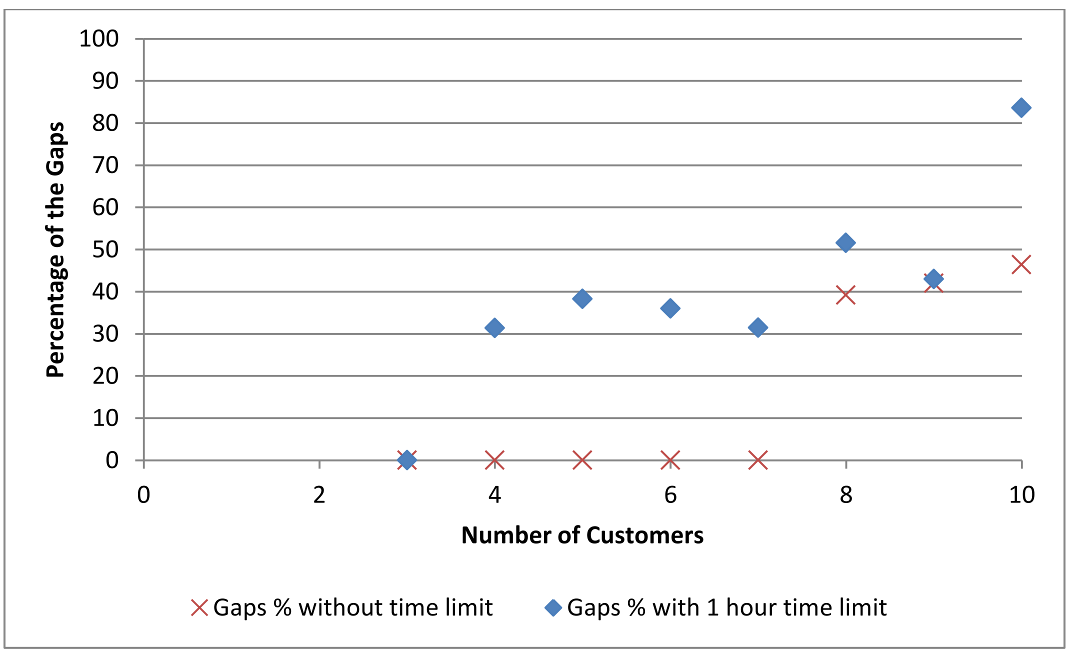

Figure 3.2: MIP Gaps with and without Time Limits for MIP Model of IPDSPPP for 8 Problems

Table 3.1: Computational Results for IPDSPPP with MIP Model for 10 Problems

Number of Customers	Planning Horizon	Number of Variables	Number of Constraints	Gap % w/o Time Limit	Soln.	Time (sec)	Gap % w/ 1hr Limit	Soln. w/ 1hr Limit
3	100	2170	4557	0	626.4	3	0	626.4
4	150	5741	12036	0	680.0	11264	31.35	680.0
5	150	9516	20217	0	580.0	37405	38.28	580.0
6	150	14635	31634	0	675.3	8565	35.97	675.3
7	150	21386	47115	0	774.5	99028	31.44	774.5
8	150	34329	63408	39.18	1036.3	79388	51.58	1049
9	150	48223	87254	42.03	1029.8	78693	43.02	1047.8
10	150	63231	120182	46.37	1099.8	76324	83.60	1326.8
15	250	262621	521172	n/a	Out of Memory	-	n/a	Out of Memory
20	250	556061	1417112	n/a	Out of Memory	-	n/a	Out of Memory

These numerical examples show us that the mixed integer programming model of the IPDSPPP problem can be solved for an optimal solution with the fleet size, vehicle type, routing sequence, delivery time schedule and plant production schedule. The MIP model sizes growth quickly when the number of customers increased. At the same time, the solving time for the optimal solution takes an approximate average 7 hours. Most important, the exactly optimal solution only can be found when there are 7 or less customers in the system under current software and hardware. Most real problems contain more than 8 customers and this issue is addressed in the next section.

Heuristic Algorithms for the IPDSPPP

Heuristic Approach Structure for the IPDSPPP

Recall that the IPDSPPP problem falls within the class of nondeterministic polynomial-time hard problems (NP-hard problem). This means that all known algorithms that define an optimal solution require exponentially increasing computational time as the number of customers increase; therefore, heuristic methods which provide approximate solutions are justified and are required for realistic sized problems. In the previous section, it was shown that exact solutions to the MIP model can only be found for situations involving 7 or fewer customers and a feasible solution could not be found for 20 customers. To extend this work to more realistic problems, several heuristic approaches are now proposed to find approximate solutions to IPDSPPP problem.

Introductory Example

It has been established that the optimal solution for the IPDSPPP problem needs to determine the production schedule and distribution schedule simultaneously, but this makes the problems extremely complex. Because of the perishable property of the product, it is clear that, in general, production of a batch of product will end relative near when distribution begins; however, production for customers located near the plant could take place in advance as long as the holding time at the plant plus the transportation time does not exceed the product lifetime. Since the production and distribution schedules are so tightly coupled, slightly changing the production sequence will dramatically affect the distribution schedule which can lead to dramatically different solutions. To illustrate the solution of IPDSPPP problem, consider the following example.

Example 3.1: Consider an IPDSPPP problem with 2 types of vehicles and 6 customers (n=6) with the following values of the model parameters: C_1=10, C_2=7, R_1=1, R_2=0.7,

B=9, H=35, r=1. Customer demands are q_1=3, q_2=3, q_3=5, q_4=2, q_5=4, q_6=3, and the travel time matrix is shown in Figures 3.3.

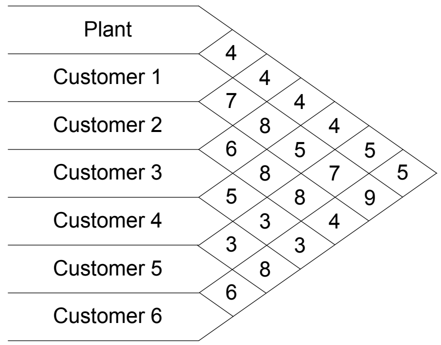

Figure 3.3: Travel Time Matrix for Example 3.1

Figures 3.4 and 3.5 illustrate a feasible solution for the Example 3.1. To facilitate discussing this solution, define the following:

Batch: It is assumed that the plant can combine the demand of one or more customers together for production as long as it does not exceed the plant's capacity. This is a batch. The production time for a batch increases linearly with the demand. The lifetime of all products in a batch begins at the same time coinciding with the time that the last product in the batch is completed. All demands in one batch will be delivered with one vehicle with one route.

Production Permutation (σ): This is the production sequence for customer orders in the plant and all orders, which represent the demand for one customer, must be produced in the order in which they appear in the production permutation. As such, a feasible production permutation for an n customer problem is an ordered list which includes numbers from 1 to n, and each number i represents the customer i. For instance, σ = <2, 4, 3, 1> is a feasible production permutation for the 4 customer problem. $\sigma(i)$ is the i^{th} customer in the production permutation, i.e. $\sigma(1)=2$. It is assumed that batches can be formed only using customer demands that are consecutive in the production permutation. To illustrate, consider the production permutation σ = <6, 3, 2, 4, 5, 1> from Example 3.1. This means that the demand for customer 6 is produced first, customer 3 second, and so forth. Further, customer 6, 3 and 2 can be combined together to be a batch. But the customer 6 and 2 are not able to be produced in one batch if the customer 3 is not included in that batch. Note that the total possible number of production permutations with n customer is $n!$.

Trip: This is a delivery route executed by a single vehicle of a specific type. It starts at the plant, delivers all customers product in the batch, and ends at the plant. The trip follows the sequence of customers in a product permutation. Hence, trips are generated to include the customers to be satisfied in the order specified by the product permutation and including the information on the required vehicle type. For example, identifying the plant as location 0, a trip for a type 1 vehicle in Example 3.1 with the production permutation σ = <6, 3, 2, 4, 5, 1> is $0 \rightarrow 3 \rightarrow 0$. Type 2 vehicle with the same route $0 \rightarrow 3 \rightarrow 0$ is considered as another trip.

Trip Delivery Time: The time duration beginning when the delivery vehicle departs the plant and ending when it delivered the product to the last customer in the trip.

Trip Travel Time: The time duration beginning when the delivery vehicle departs the plant and ending when it finishes the route by returning to the plant. That is, it is the trip delivery time plus the travel time from the last customer to the plant.

Feasible Trip: A feasible trip is one that meets all of the requirements including the capacity constraint, lifetime constraint and delivery rules. That is, if the amount of product required for a trip exceeds the vehicle capacity or the trip delivery time is longer than the product lifetime, the trip is infeasible.

Feasible Trip Set (φ): The set of all feasible trips from a given production permutation. Table 3.2 provides the feasible trip set for Example 3.1 with production permutation σ = <6, 3, 2, 4, 5, 1>.

Table 3.2: Feasible Trip Set for Example 3.1 with Production Permutation σ = <6, 3, 2, 4, 5, 1>

Trip name	Included customers	Delivery Route	Vehicle Type	Vehicle Load	Delivery Time
A1	6	0→6→0	1	3	5
B1	6,3	0→6→3→0	1	8	8
C1	3	0→3→0	1	5	4
D1	2	0→2→0	1	3	4
E1	4	0→4→0	1	2	4
F1	4,5	0→4→5→0	1	6	7
G1	5	0→5→0	1	4	5
H1	1	0→1→0	1	3	4
A2	6	0→6→0	2	3	5
C2	3	0→3→0	2	5	4
D2	2	0→2→0	2	3	4
E2	4	0→4→0	2	2	4
F2	4,5	0→4→5→0	2	6	7
G2	5	0→5→0	2	4	5
H2	1	0→1→0	2	3	4

Trail (ρ): A list of feasible trips from the feasible trip set with a fixed order that satisfies all demands within the time horizon. The trail must contain trips that cover all customers once and only once. At the same time, the customer sequence delivered by trips in the trail match the customer sequence in the production permutation. To illustrate, a trail for Example 3.1 is ρ = <B1, D2, F1, H2> because B1 covers customers 6 and 3, D2 covers customer 2, F1 covers customers 4 and 5, and H2 covers customer 1. This can be seen in Table 3.2. Hence, this trail covers every customer once and only once and maintains the same customer sequence as in the production permutation, σ = <6, 3, 2, 4, 5, 1>.

In closing this discussion about Example 3.1, note that Figures 3.4 and 3.5 illustrate a feasible solution. The production permutation for the solution is σ = <6, 3, 2, 4, 5, 1>. 6 customers are delivered using 4 trips in the trail ρ = <B1, D2, F1, H2>. This trail uses 2 vehicles (one Type 1 vehicle and one Type 2 vehicle) and each vehicle makes 2 trips within the time horizon. The batch production schedule along with the distribution time schedule and trip delivery routes are presented for this feasible solution in Figure 3.5. The solution includes the production time schedule for each batch and its sequence, the delivery time schedule for each vehicle, their delivery trips in the horizon, the routing sequence in each trip and the type for each delivery vehicle.

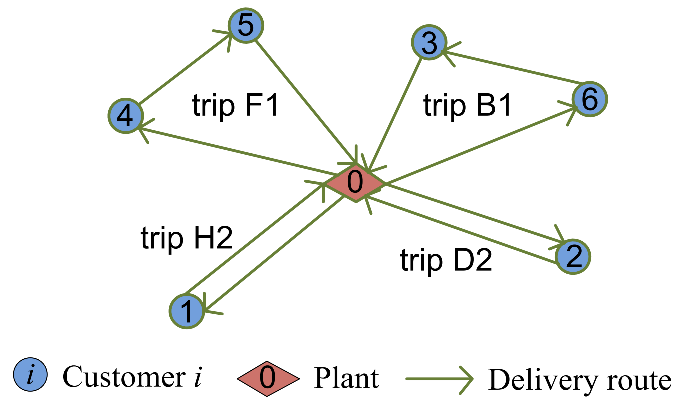

Figure 3.4: Customer Locations and a Distribution Routes Plan for Example 3.1

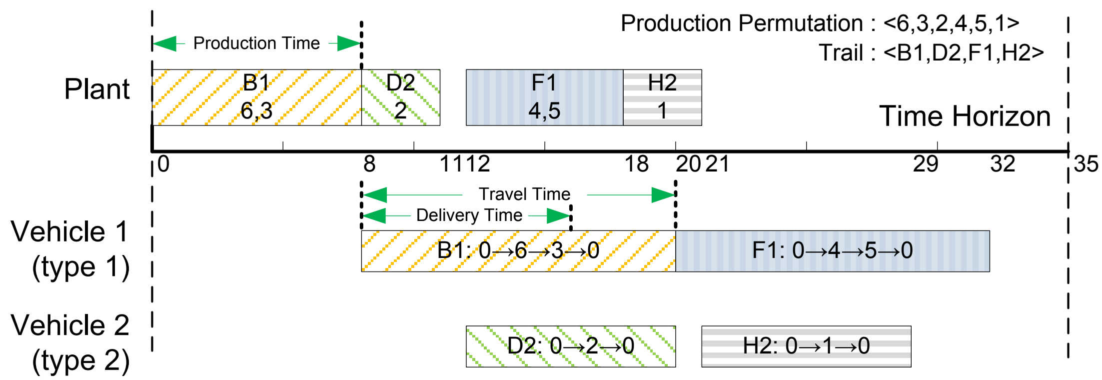

Figure 3.5: A Feasible Solution for Example 3.1

Notation and structure of the heuristic algorithm

Figure 3.5 illustrates the important fact that feasible solutions to IPDSPPP have tightly coupled production and delivery schedules. Any change in the production schedule can not only affects the production batch size but also impacts the delivery route and dispatch time schedule, and eventually affect the final transportation cost. This observation motivates the main idea for the heuristic algorithms; namely, first create the initial production schedule, then generate the distribution routes based on the initial production sequence, and finally generate the distribution schedule and refine the production schedule simultaneously. This approach is translated into a general structure that will be applied to develop heuristics that find approximate solutions for the IPDSPPP problem. Figure 3.6 shows the basic hierarchy of decisions in the heuristic algorithm for IPDSPPP problem.

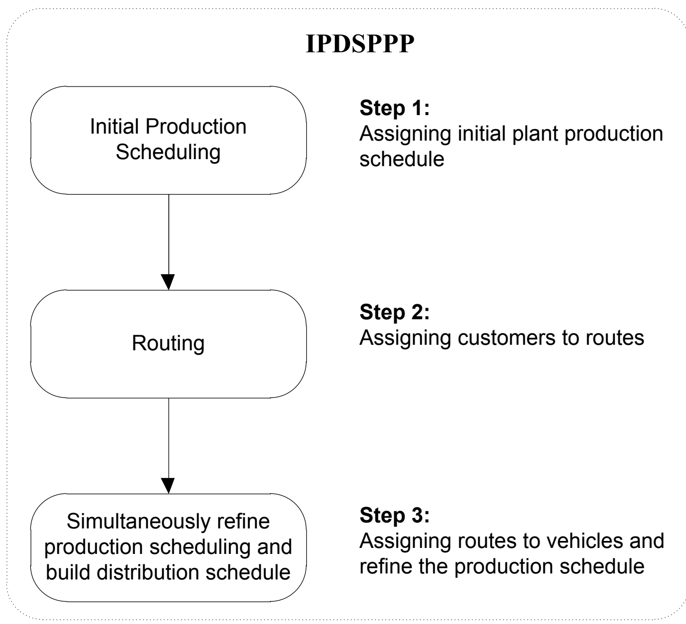

Figure 3.6: The Basic Hierarchy of Decisions in the Heuristic Algorithm of IPDSPPP

To implement the decision hierarchy shown in Figure 3.6, two elements need to be carefully designed:

1) As shown in Figure 3.6, the solution of the IPDSPPP is generated from the production schedule. Hence, a search technique is needed in step 1 that effectively explores the possible outcome space of the initial production schedules to find improved solutions.

2) A way to interpret the initial production schedule to a result schedule (steps 2 and 3) is also required. Recall the feasible solution in Example 3.1, a feasible solution, in general, contains more information than just a production schedule. The initial production schedule must conform to all rules and constraints before it becomes a feasible solution. So, the way to interpret the initial production schedule is also very important.

From these ideas, a detailed heuristic algorithm decision flow chart for the IPDSPPP problem is shown in Figure 3.7. A production permutation is used as the initial production schedule. The classical approach "routes first, cluster second" is used to solve the distribution schedule.

As shown in the Figure 3.7, the heuristic decisions are divided into 6 steps that can be grouped into 2 large parts: 1) the interpret part and 2) the random improvement part. The random improvement part focuses on the initial production scheduling decisions and it includes step 1, 5, 6 and the loop structure. The Interpret part focuses on the routing decision, refining the production schedule, and building distribution schedule decisions. It includes step 2, 3 and 4.

The heuristic works in the following sequence. First, the random improvement part randomly generates a production permutation and passes it to the interpret part. By passing through steps 2, 3 and 4 in the interpret part, the production permutation is interpreted into a solution, possibly infeasible, for the IPDSPPP problem. Then the solution is sent back to the random improvement part to check the feasibility. The feasible solution will be compared with the current best solution and saved if it is better.

Then a new production permutation is generated to start another loop if the stop criteria had not been met.

The next section introduces more details about the random improvement part and the interpret part.

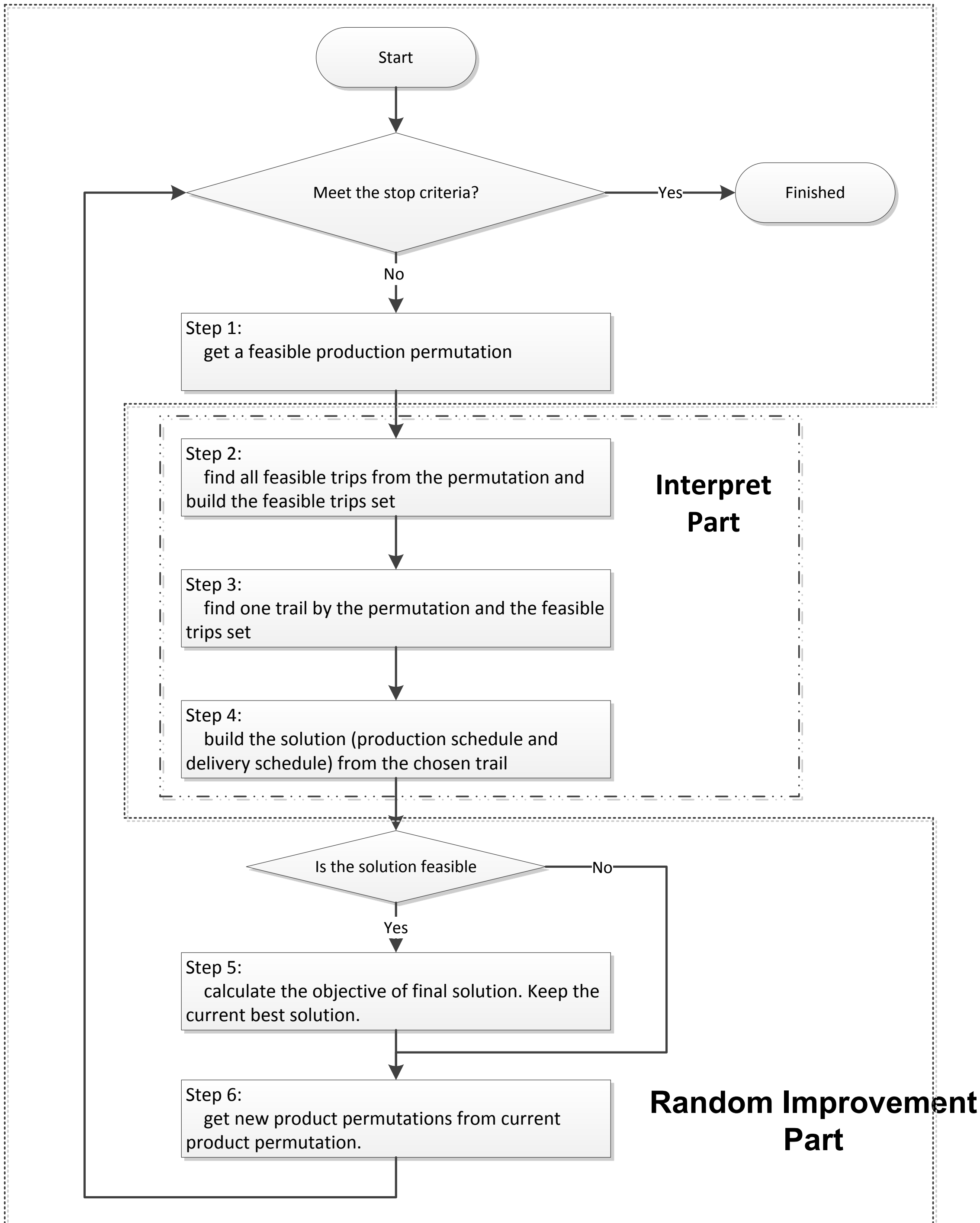

Figure 3.7: Heuristic Algorithm Decision Flow Chart for the IPDSPPP

Random Improvement Part with Simulated Annealing Algorithm

The function of the random improvement part is to search for production permutations that can produce better feasible solutions by the interpretation process progresses. As mentioned previously, the total number of production permutations for an *n* customer system is *n*!, so complete enumeration would also require a nondeterministic polynomial amount of time. Hence, a search technique is justified to find a production permutation that might lead to a better solution from a previous iteration. In this research, a simulated annealing (SA) search methodology is used.

SA is a probabilistic based search technique inspired from the annealing of metals which uses heating followed by controlled cooling to improve some of the material's properties such as strength and hardness. SA uses the concepts of states, temperature, energy, and cooling function to mimic some of the physical process of annealing. In general, the SA search wants to make a random movement searching for a minimum system energy level. At the very beginning, the system will have an initial state and temperature (T). Each movement will change the system from current state to another state. At the same time, each movement will consume heat and reduce the system temperature (T) by the cooling schedule $T(k)$, where k is the total number of movements from the initial state. The whole search is stopped when the temperature is lower than then frozen temperature (T_f). The current state (s) saves the current state of the system. The energy function maps each state (s) to a system energy level (e), $e=E(s)$. The objective of the SA algorithm is to find the minimum system energy level. To do so, the SA algorithm will first make a random movement to find a neighbor state (s') from the current state (s) by using the neighborhood function, $s' = N(s)$. A new system energy level (e'), $e'=E(s')$ is computed and the difference of two system energy levels (ΔE) is calculated, $\Delta E=e'-e$. If the system energy level is reduced at the new state, i.e. $\Delta E<0$, the system will accept the movement and save the new neighbor state (s') as the current state (s). If the system energy level is not reduced, i.e. $\Delta E>0$, there is still a possibility that the movement is accepted. This is controlled by $P(e,e',T)$, the transition probability function which is a special feature of SA. Since the objective of the SA search is to seek the minimum system energy level, the transition probability allows the system accept a "worse direction", that is, a movement that increases the system energy. This strategy helps the search avoid getting stuck at the local optimum. Although the immediate move with this method might not always beneficial, but it provides a chance to get better solution than current local optimum. For example, Figure 3.8 illustrates this idea because if the SA algorithm only accepts a movement from the current state to lower energy state such as the movement from s to s'', it will always reject the movement from s to s' which provides the chance to reach the global optimum.

Figure 3.9 is the pseudo code for the general concept of simulated annealing search algorithm, and the functions might vary in different implantations.

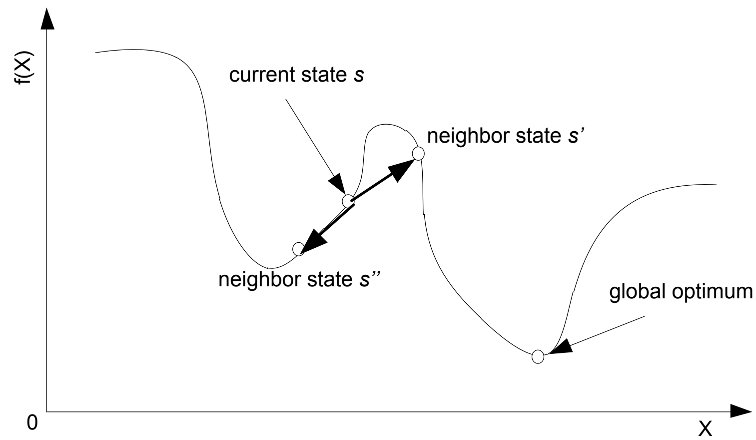

Figure 3.8: Example of State Movement for SA Algorithm

Define an initial temperature T, a frozen temperature T_f.
Define a cooling schedule $T(k)$.
Compute the objective value e=$E(s)$ for current solution s. e.g. direct transportation
$k = 0$
while ($T > T_f$) {
    randomly get one neighborhood s'=$N(s)$ from current solution s
    Compute the energy e'=$E(s')$ for the solution and calculate the ΔE= e' - e
    If ($\Delta E < 0$)
    Then
        Accept the new solution s', do the transition
    Else
        Accept the new solution s' with probability $P(e,e',T)$
    $k = k + 1$
    $T = T(k)$
}

Figure 3.9: Pseudo Code of the Simulated Annealing Algorithm

In this research, system energy is the total system transportation cost. The production permutation encodes the system state. The interpret part is the calculation function for the system energy. In every iteration, a new production permutation will be generated from current production permutation. After the interpret part, the new production permutation will be accepted or discarded. For sure, if the new production permutation can create a feasible schedule and provide lower transportation cost, it will be accept immediately. Also, if the new production permutation leads to a infeasible schedule, the new production permutation will be discarded. Furthermore, if the new production permutation leads to a feasible schedule but increase the system total transportation cost, it will be accept will probability of $P(e,e',T)$ and be discarded with probability of 1 - $P(e,e',T)$. The accepted production permutation will replace the current production permutation, while the discarded production permutation is not been recorded. The new production permutation is always generated from current production permutation by randomly chosen neighborhood function.

A neighbor of the current production permutation σ is a new production permutation σ' which has a minor difference from the current production permutation in some certain way, such as a different sequence of customer sequence. The neighborhood function, $N(\cdot)$, is the function that randomly generates a new production permutation σ' from current production permutation σ, $\sigma' = N(\sigma)$. In this research, two neighborhood functions are used to identify neighbors: 1) swap, and (2) move. The swap function generates a neighbor of the current production permutation by swapping two customers in the original production permutation. The move function generates a neighbor of current production permutation by moving the i^{th} position's customer to the j^{th} position. The customers between i^{th} and j^{th} position will move forward or backward 1 position to fill the vacant positions created by moving the i^{th} customer. To ensure that a neighbor is generated randomly, the selection of neighborhood functions and relative parameters for each movement is random chosen. Figure 3.10 displays two neighbors for the production permutation σ = <6, 3, 2, 4, 5, 1>, one generated by each method.

$$\sigma = <6, \mathbf{3}, 2, 4, \mathbf{5}, 1> \xrightarrow{\text{swap } 2^{nd} \text{ and } 5^{th} \text{ customer}} \sigma' = <6, \mathbf{5}, 2, 4, \mathbf{3}, 1>$$

$$\sigma = <6, 3, 2, 4, \mathbf{5}, 1> \xrightarrow{\text{move } 5^{th} \text{ customer to } 2^{nd} \text{ position}} \sigma'' = <6, \mathbf{5}, 3, 2, 4, 1>$$

Figure 3.10: Two Neighbors of Production Permutation σ = <6, 3, 2, 4, 5, 1>

Interpret Part with Different Heuristic Algorithms

The previous sections focused on the random improvement part use the SA algorithm and now attention is turned to the interpret part that converts the production permutation into a feasible solution including both the production schedule for the plant and distribution schedule for the delivery fleet.

In general, all heuristics discussed in this section follow the decision flow chart shown in Figure 3.11. First of all, the feasible trips set will be built based on the production permutation. In this step, all customers will be divided into different trips. In each trip, the product lifetime constraint is satisfied, which means that if the product is delivered immediately after finishing production, every customer can get the qualified product in time. Also, the vehicle capacity constraint is guaranteed in each trip. Then, a trail will be selected which includes a group of feasible trips from the feasible trip set. This group of trips in the trail must cover every customer once and only once. Also, the production schedule for each trip must follow the sequence in production permutation. After that, the number of vehicles and the type for each will be estimated and all trips in the trail will be packed into these vehicles. By considering the basic rules, such as batch production must be finished before delivery can commence, the distribution routing and the production schedule are built simultaneously. Since we need to finish all delivery before the fixed time horizon, a refine process is followed to make the production and distribution schedule more compact without violating the product lifetime constraint. After that, a solution of the problem is created and the feasibility of the solution will be checked. A feasible solution means that all customer demand is delivered to customer place before product lifetime expired and all delivery vehicles return to the plant before the end of time horizon. If a feasible solution cannot be found with current estimated number of vehicles, the algorithm either increases the number of vehicles by 1 or indicates that there is no feasible solution. The latter is deduced by checking if the current number of vehicles is equal to n. If it is, then no feasible solution exits for the production permutation because n customers cannot be delivered by n vehicles. Once a feasible solution had been find or no feasible solution had been conclude, the interpret part is finished. The result (feasible/Non-feasible solution) will be passed to the random improvement part.

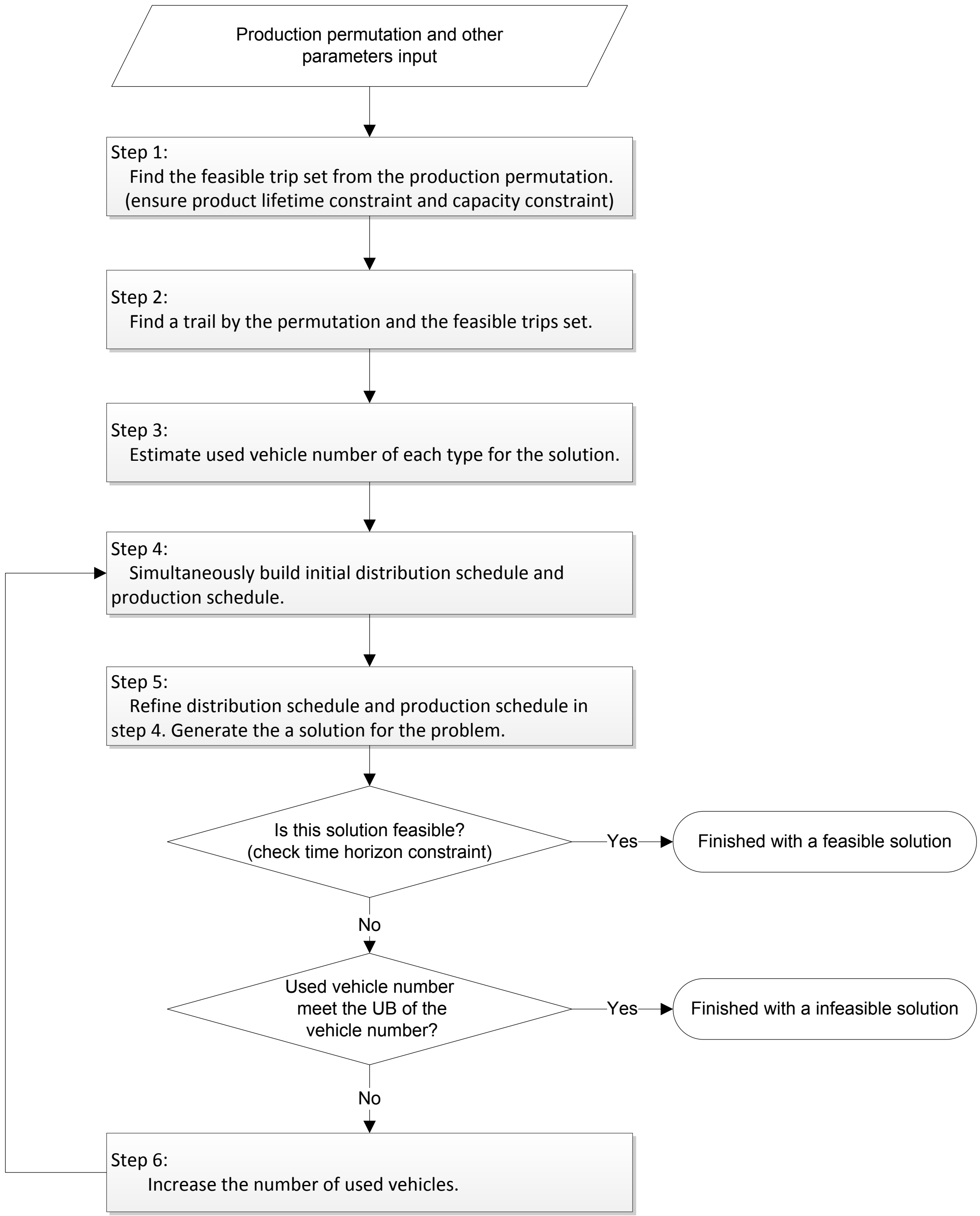

Figure 3.11: Basic Hierarchy of the Decisions in Interpret Part

1. Find the feasible trip set form the production permutation

This is the first step in the heuristic. Recall the definitions; the production permutation is the production sequence for all customer orders in the plant. The feasible trip is a delivery route executed by a single vehicle of a specific type, follows the sequence of customers in a product permutation, and meets all requirement of capacity constraint, lifetime constraint and delivery rules. As such, the production permutation limits the delivery sequence for the trip; the lifetime constraint limits the length of the trip; the capacity constraint limits the load of the trip; and the delivery rule (i.e. direct delivery) limits some other properties of the trip. To better illustrate the progress, Example 3.2 is provided.

Example 3.2: Consider a 3 customer problem with q_1=1, q_2=2, q_3=2, customer and plant locations are shown in Figure 3.12, B=5, and H =8, two types of vehicle with different capacity C_1=4, C_2=3, variable cost R_1 =1, R_2=0.9. For this example, we only allowed direct delivery. Table 3.3 shows all trips for the production permutation σ =<3, 2, 1>.

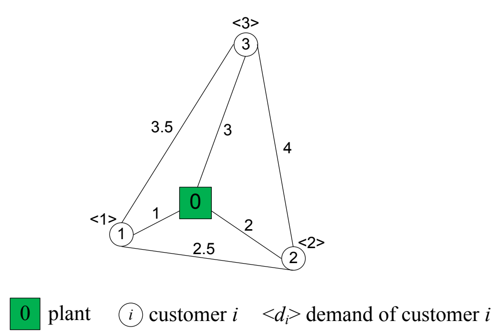

Figure 3.12: Customers and Plant Location for Example 3.2

Table 3.3: All Trips List for Example 3.2

Trip Name	Included customers	Delivery Route	Vehicle Type	Vehicle Load	Delivery Time
A1	3	$0\to3\to0$	1	2	3
B1	2	$0\to2\to0$	1	2	2
C1	1	$0\to1\to0$	1	1	1
D1	3,2	$0\to3\boldsymbol{\to}2\to0^{*}$	1	4	7^{***}
E1	2,1	$0\to2\boldsymbol{\to}1\to0^{*}$	1	3	4.5
F1	3,2,1	$0\to3\to2\boldsymbol{\to}1\to0^{*}$	1	5^{**}	9.5^{***}
A2	3	$0\to3\to0$	2	3	3
B2	2	$0\to2\to0$	2	2	2
C2	1	$0\to1\to0$	2	1	1
D2	3,2	$0\to3\boldsymbol{\to}2\to0^{*}$	2	4^{**}	7^{***}
E2	2,1	$0\to2\boldsymbol{\to}1\to0^{*}$	2	3	4.5
F2	3,2,1	$0\to3\to2\boldsymbol{\to}1\to0^{*}$	2	5^{**}	9.5^{***}

Note: * indicates delivery rule violation, ** indicates capacity constraint violation, *** indicates lifetime constraint violation.

Table 3.4: Feasible Trip Set for Example 3.2 with Production Permutation σ =<3, 2, 1>

Trip Name	Included customers	Delivery Route	Vehicle Type	Vehicle Load	Delivery Time
A1	3	$0\to3\to0$	1	3	3
B1	2	$0\to2\to0$	1	2	2
C1	1	$0\to1\to0$	1	1	1
A2	3	$0\to3\to0$	2	3	3
B2	2	$0\to2\to0$	2	2	2
C2	1	$0\to1\to0$	2	1	1

As shown in Table 3.3, the trips D1, E1, F1, D2, E2, F2 violate the delivery rule; trips F1, D2, F2 violate the capacity constraint and trips D1, F1, D2, F2 violate the lifetime constraint. By removing all infeasible trips from the list, the feasible trip set for production permutation σ =<3, 2, 1> for Example 3.2 is shown in Table 3.4.

2. Find a trail by the production permutation and the feasible trip set

Once the feasible trip set is built, the next step is to find a trail. Recall previous description, a trail contains several trips chosen from the feasible trip set that cover each customer once and only once. At the same time, the customer sequence delivered by trips in the trail match the customer sequence in the production permutation. The purpose to find a trail is to choose a group of feasible trips to perform the distribution function. Once a trail is determined, the total number of trips used in the distribution plan is fixed which implies that the total system variable cost is determined for the current distribution plan. For example, a feasible trail for Example 3.2 is ρ = <A2, B1, C1>, these 3 trips cover all 3 customers in sequence <3, 2, 1> which matches the sequence of production permutation σ =<3, 2, 1>. Since the trail is determined, the trips used for distribution is determined (trip A2, B1 and C1), and the total variable cost for the distribution plan can be calculated as follows:

Total variable cost = variable cost for trip A2 + variable cost for trip B1

+ variable cost for trip C1

= 6*0.9 + 4*1 + 2*1 = 11.4

3. Estimate used vehicle number of each type for the solution

From previous step, it provides all trips used for the distribution. But the total number of used vehicle for each type is not determined. The number of vehicles used will determine the total system fixed cost, the fewer vehicles used in each type the less total fixed cost will be. The simplest way to estimate the used vehicle number is to assign one vehicle for each trip. In this way, every vehicle will delivery only one trip during the time horizon. Although this method is easy to apply, none vehicle is reused in the time horizon and remain lots of idle time for every vehicle. In this research, a trial and error approach is used to find the used vehicle number for each type. One reason for using the trial and error approach is that the search is not extensive and can be well defined by a lower bound (*LB*) and upper bound (*UB*). Note that each trip is associated with a fixed type of vehicle. Tables 3.3 and 3.4 illustrated this. Trips are first divided into subsets according to the type of vehicle associated with them. If there are *s* types of vehicles, trips are divided into *s* subset. The *UB* and *LB* are then generated for each subset *i* as follows:

UB_i = number of *trips* in the subset *i*, and

$$LB_i = \left\lceil \frac{\textit{sum travel time of all trips using type i vehicle}}{\textit{length of planning horizon}} \right\rceil$$

The feasible trail ρ = <A2, B1, C1> for Example 3.2 with production permutation σ =<3, 2, 1> uses three trips: 1 trip uses type 1 vehicle and 2 trips use type 2 vehicle. So, the *LB* and *UB* for this trail can be calculated, $LB_1 = \left\lceil \frac{2+4}{8} \right\rceil = 1$, UB_1=2 and $LB_2 = \left\lceil \frac{6}{8} \right\rceil = 1$, $UB_2 = 1$.

4. Simultaneously build initial distribution schedule and production schedule.

Though the trail information and number of vehicles to be used provide the general framework of a solution, the production and delivery schedule still need to be determined to ensure that there is no conflict leading to infeasibility. Specifically, infeasibility can still be caused by bad coordination between production schedule and distribution schedules, such as trip starting delivery before its production finished, to start delivery causing product to expire. This part of the algorithm is designed to prevent this if it is possible.

Once a trip is defined several items follow naturally. One is that the plant must complete production of the batch to be delivered prior to the trip beginning. In addition, since it is more cost effective for a vehicle to make multiple trips rather than multiple vehicles making one trip each, it is reasonable that production of the product for the "next" trip to be delivered by a vehicle occur while the current trip is underway if that is possible. Further, it is desirable for delivery to begin immediately after production or close to this time so that the maximum time possible is reserved for the delivery before the product lifetime expires. This problem has strong parallels to the two-machine, no-wait, flow shop scheduling problem with limited makespan. Here, makespan is the elapsed time between starting production of a batch and completing production of that batch. No-wait means jobs are not allowed to wait between two successive machines which implies that the starting time of a job at the first machine has to be delayed to ensure that the job can go through the flow shop without having to wait for the second machine. Think of the production process as analogous to the first workstation in a single, serial production line. The delivery process is the second workstation with multiple vehicles used to satisfy the distribution function. All delivery vehicles share the same plant and must guarantee they can finish their delivery within the time horizon to maintain feasibility of the solution. The trail that is generated in previous steps not only gives the trips production time and delivery routes, it also provides the production sequence for each trip in the schedule. By applying the no-wait rule and the trips production sequence based on the trail, the initial integrated production schedule and distribution schedule can be built. For convenience, name the initial integrated production schedule and distribution schedule as no-wait schedule. Figure 3.13 illustrate an example for the no-wait schedule. There are 5 trips in the trail = <D, A, E, B, C>, all trips are using type 1 vehicle and the estimated used vehicle number for type 1 vehicle LB_1=3. Lifetime B=8 and time horizon H=14. Each trip is immediately delivered after production is finished.

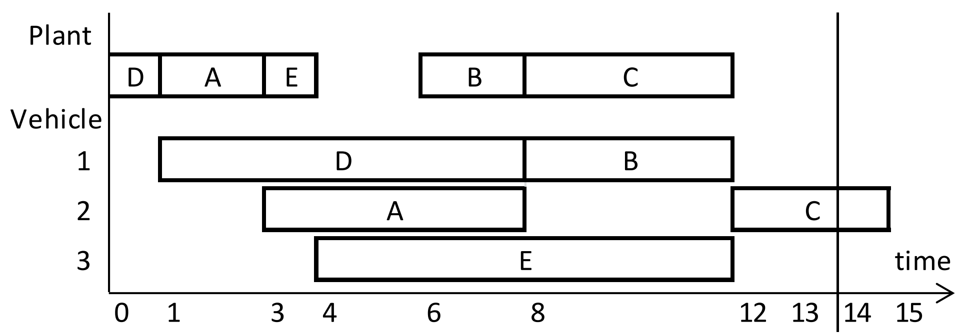

Figure 3.13: Gantt Chart for No-Wait Schedule, B=8, H=14

5. Refine distribution schedule and production schedule.

The no-wait schedule ensures that all products are delivered immediately after production is completed without holding them in inventory. This guarantees every trip delivers the product to meet the customer's demand before the lifetime expires provided the problem is feasible in the first place. The solution, however, can still be infeasible because all vehicles do not have time to return to the plant before the end of the time horizon. This is illustrated in Figure 3.13 where it can be seen that vehicle 2 cannot make the delivery and return to the plant within the time horizon. A more careful examination of Figure 3.13 suggests that a solution to the problem might be available. Notice that missing the delivery is actually caused by the fact that vehicle 2 must wait from time period 8 to 12 for its second batch to be produced. This wait was created by correct application of the no-wait rules to batch B. By itself, this is of limited help; however, Figure 3.13 also reveals that batch B has a short delivery time so its production might be able to occur earlier in the schedule. If this occurred, then it would be held in inventory for a short period of time but feasibility must be maintained so delivery of batch B to all its customers would still need to occur before the lifetime expires. Obviously, the cascading influence this has on the initial problem of batch C is that if feasibly can be maintained and production of batch B can be moved earlier by an amount that allows batch C production to complete in time for that delivery to be made, this trail is made feasible. This basic idea is formalized in this algorithm where the opportunity to "compress" the no-wait production schedule and distribution schedule is investigated to see if it will reduce the makespan of the entire production permutation and convert an infeasible schedule into a feasible one. Later, we will refer the refine the no-wait schedule as compressed schedule. Figure 3.14 is the compressed schedule for Figure 3.13 that graphically shows the adjustments do create a feasible solution.

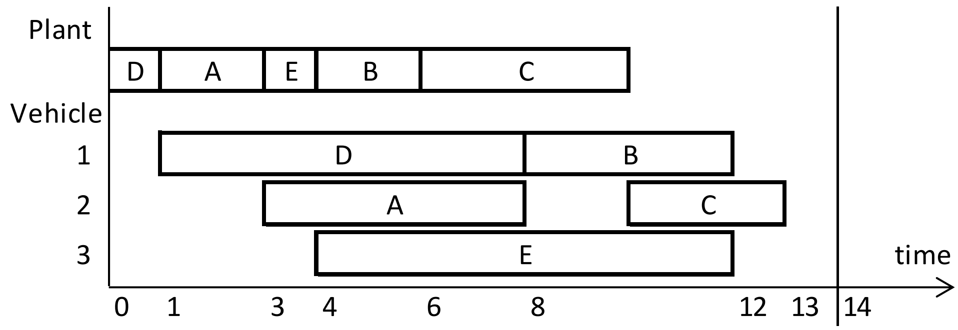

Figure 3.14: Gantt Chart, compressed schedule for Figure 3.12, *B*=8, *H*=14

Since the no-wait schedule initiates delivery immediately when production of a batch is completed, the compression process should start with the production schedule because, by definition, there are no gaps in the distribution schedule. If a production schedule is compressed, it means that the trip batch completion time has been moved earlier in time. This translates into the opportunity for the departure time of the trip carrying that batch to also be shifted to an early time to eliminate the induced gap of idle time created by earlier production. On the other hand, compression is not without potential problems. After the compression, a situation can be created in which the batch that initiates the production is so early that, after holding the product in inventory until its trip commences, delivery cannot be made before the lifetime expires. Clearly, creating a situation where one deadline is missed to resolve missing another is not acceptable. To address this, recall that the trip delivery time is the time duration from when the vehicle departs the plant until the time the product is delivered to the last customer in the trip and the trip travel time is the sum of trip delivery time and the time it takes the truck to return to the plant from the last customer in the trip. Using these concepts, *trip maximum wait* *time* is defined as product lifetime minus trip delivery time. The idea, then, is that after any compression, the waiting time for each trip must less or equal to its trip maximum wait time. Otherwise, the trip start production time must to be postponed, until a new schedule is generated that meets this criteria.

The algorithm used to compress the no-wait schedule is now presented. To begin, define the following:

i: Numbers the batches in a production schedule with i=1 designating the first batch produced. This also numbers the trips carrying the batches. The i^{th} trip in the production schedule is represented as trip i.

u_i: trip i's batch starting production time

v_i: trip i's batch starting distribution time

d_i: trip delivery time for the trip i

t_i : trip travel time for the trip i

p_i : production time for the trip i

a_i: the trip maximum wait time for trip i

K: the gap in the production schedule

L: the gap in the distribution schedule

$\Phi(i)$: a function that identifies the previous trip's identification number in the distribution schedule for trip i ,(e.g. in Figure 3.13, $\Phi(C)$=A).

Using these definitions, the steps in the algorithm are now presented.

Step 1: set i = number of trucks + 1

Step 2: if $u_i > u_{i-1} + p_{i-1}$ then go to step 3, else go to step 7.

Step 3: get the production gap $K = u_i - (u_{i-1} + p_{i-1})$, find previous trip in distribution schedule $j = \Phi(i)$, and get the distribution gap $L = v_i - (v_j + t_j)$. And go to step 4.

Step 4: if $K > L$ then go to step 5 else go to step 6.

Step 5: compress the production schedule, $u_i = u_i - L - Min(K - L, a_i)$, and compress the distribution schedule $v_i = v_i - L$. go to step 7

Step 6: compress the production schedule, $u_i = u_i - K$, and distribution schedule $v_i = v_i - K$. go to step 7

Step 7: $i = i + 1$, if $i <= m$ then go to step 2, else compression finished.

This strategy can be translated into the following pseudo code:

```
i = 2
While (i <= m) {
    If u_i > u_{i-1} + p_{i-1}   then{
        K = u_i – (u_{i-1} + p_{i-1})
        j = e(i)
        L = v_i – (v_j + t_j)
        If (K > L)  then {
            u_i = u_i – L – min(K – L, a_i)
            v_i = v_i – L
        }
        else{
            u_i = u_i – K
            v_i = v_i – K
        }
    }
    i = i + 1
}
```

Figure 3.15: Pseudo Code to Compress the No-Wait Schedule

6. Check the feasibility of the solution

After compressing the no-wait solution, a solution for the IPDSPPP problem is generated by the input production permutation. At this time, feasibility of the solution is still not guaranteed in general because everything has been based on an estimate of the number of vehicle for each type. So, in this step, a feasibility check needs to check whether every vehicle finishes all trips within the time horizon or not. For example, the Figure 3.14 displays a feasible solution while Figure 3.13 displays an infeasible solution.

If the solution is feasible, the used vehicle number for each type is determined. At the same time, the integrated production and distribution schedule is built which implies a feasible solution for the input product permutation is created. Otherwise, the heuristic needs to increase the number of vehicles used to find a feasible solution for the given product permutation.

7. Increase the number of used vehicles

Recall the LB definition for the number of vehicle of each type

$$LB_i = \left\lceil \frac{sum\ travel\ time\ of\ \ all\ trips\ using\ type\ i\ vehicle}{length\ of\ \ planning\ horizon} \right\rceil,$$

It might have lower estimation with following reasons: 1) did not include the idle time for the vehicle. This LB estimation is assumed that there is no idle time for the vehicle. Actually, this is not guaranteed. Such as shown in Figure 3.13, there is an idle time for vehicle 2 between delivery trip A and trip C. This can cause the lower estimation for the LB. 2) Might split the trips. For instance, suppose there are 3 trips A, B, C using type 1 vehicle with travel time 4, 4, 4. The time horizon H=6. According to the formula,

LB_1=(4+4+4)/6=2. With this estimation, 2 vehicles both have 2 unit times left when they finish the first trip delivery, and the last trip will be split delivered by these 2 vehicles.

It is noted that the sequence of a trip in a trail has an important impact on the feasibility of the solution. Suppose a problem as shown in Figure 3.16, the trail = <A, B, C> requires 3 trips to build the solution and all trips are delivered by the same type of vehicle. The overlap problem is seen in Figure 3.17 where the only feasible sequence is to use the three trips in the order A, B and C. Even with the same trips and the same vehicles, one sequence of the trips in the trail is feasible and one is not.

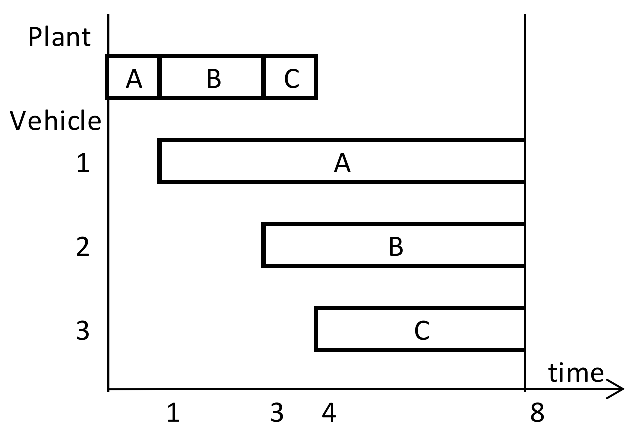

Figure 3.16: Solution with Trail = <A, B, C>

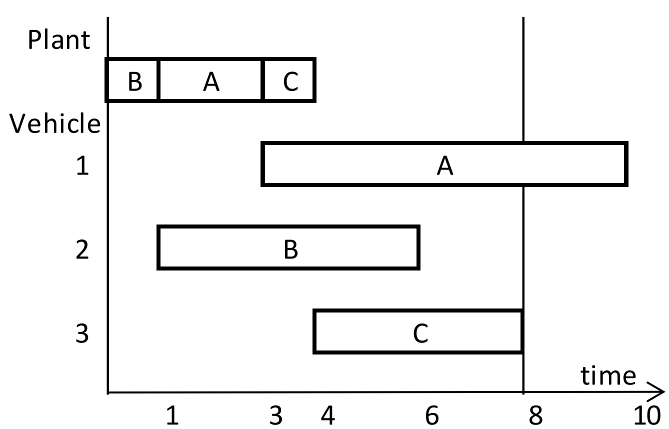

Figure 3.17: Solution with Trail = <B, A, C>

For situations where the infeasibility is cause by sequence like the one in Figure 3.17, adding more vehicles will not make it feasible. On the other hand, if the infeasibility of the solution is caused by a wrong estimation of the lower bound for each type of vehicle, such as illustrated in Figure 3.18, increasing the number of vehicles could fix the infeasibility problem as shown in Figure 3.19.

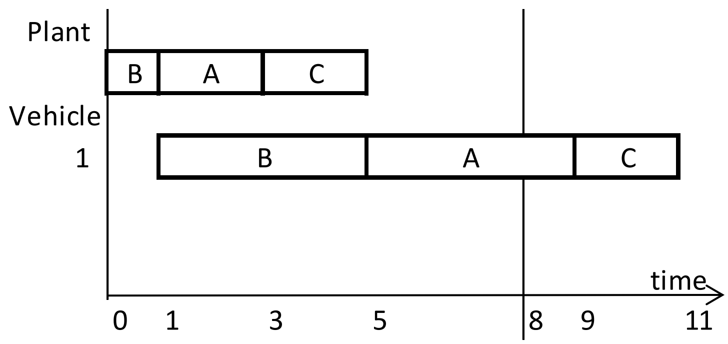

Figure 3.18: Infeasible Solution with Trail = <B, A, C>

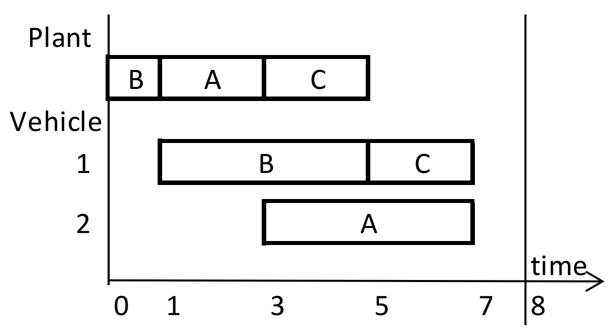

Figure 3.19: Feasible Solution with Trail = <B, A, C>

After previous steps, the interpret part could provide a solution based on the given production permutation (either feasible such as Figure 3.16 or infeasible such as Figure 3.17). The next section will discuss some detailed heuristic based on the described heuristic structure. Each of the specific heuristics that are now described take different approaches in some of these steps. Each has positive and negative features and there is not one algorithm that is best in all situations. The first heuristic is presented beginning with the simplest one and as will be seen, each subsequent heuristic improves a weakness found in one or more of the previous ones.

Direct Delivery with the Just Fit Vehicle Rule (H1)

The first heuristic method proposed is the direct delivery with just fit vehicle rule (H1). This heuristic allows a solution to be found quickly by dramatically reducing the number of possible choices for the decision maker using a prescriptive approach. In H1, all customer demands are delivered directly from the plant (direct delivery) using a vehicle with minimum possible capacity (just fit vehicle). All other options that could reduce cost, like using vehicles of larger capacity that would allow multiple stops, are not considered. Applying these two rules allows the vehicle type and routes to be quickly determined for each customer delivery. The implementation of this algorithm is rather straightforward using the overall strategy previously discussed.

1. Generating a feasible trip set

Here, trips are feasible only when they can both successfully deliver product to the customer satisfying all constraints and use the smallest capacity vehicle that can achieve this. Furthermore, it needs to satisfy all other delivery rules.

To implement the direct delivery rule, the decision maker only needs to know each customer's demand to determine vehicle type and location to check for feasibility, so that all feasible trips can be generated. Example 3.3 is used for the illustration. It has the same information with Example 3.1 except time horizon.

Example 3.3: Considers an IPDSPPP problem with 2 types of vehicles and 6 customers (n=6) with the following values of the model parameters: C_1=10, C_2=7, R_1=1, R_2=0.7, F_1 =120, F_2=100, B=9, H=31, r=1. Customer demands are q_1=3, q_2=3, q_3=5, q_4=2, q_5=4, q_6=3. Refer to Example 3.1 for plant and customer locations.

The direct delivery rule is limited to trips that can deliver one customer in its route. Table 3.5 shows all feasible trips with production permutation $\sigma = <6, 3, 2, 4, 5, 1>$ using direct delivery.

The just fit vehicle rule uses vehicles with a capacity just sufficient to carry the customer's demand; hence, the algorithm selects the vehicle with the smallest capacity from those that are feasible for each customer. For instance, suppose the demand of customer i is 11, q_i = 11, and there are three different types of vehicles with the capacities, C_1 = 8, C_2 =12, C_3 =15. According to the just fit rule, one type 2 vehicle will be used because it has the minimum acceptable capacity of 12. Table 3.6 shows the feasible trip set for the Example 3.3 using the H1 algorithm.

Table 3.5: Feasible Trip Set for Example 3.3 with Production Permutation σ = <6, 3, 2, 4, 5, 1> Using Direct Delivery

Trip name	Route	Vehicle Type	Load	Delivery Time	Travel Time
A1	0→6→0	1	3	5	10
C1	0→3→0	1	5	4	8
D1	0→2→0	1	3	4	8
E1	0→4→0	1	2	4	8
G1	0→5→0	1	4	5	10
H1	0→1→0	1	3	4	8
A2	0→6→0	2	3	5	10
C2	0→3→0	2	5	4	8
D2	0→2→0	2	3	4	8
E2	0→4→0	2	2	4	8
G2	0→5→0	2	4	5	10
H2	0→1→0	2	3	4	8

Table 3.6: Feasible Trip Set for Example 3.3 with Production Permutation σ = <6, 3, 2, 4, 5, 1> Using H1 Algorithm

Trip name	Route	Vehicle Type	Load	Delivery Time	Travel Time
A2	0→6→0	2	3	5	10
C2	0→3→0	2	5	4	8
D2	0→2→0	2	3	4	8
E2	0→4→0	2	2	4	8
G2	0→5→0	2	4	5	10
H2	0→1→0	2	3	4	8

2. Generating a trail

To get the trail, the production permutation is partitioned into feasible trips. Because of the direct delivery and just fit rule, feasibility is guaranteed and there is only one feasible trail. Using the H1 method yields the feasible trail for this example

ρ = <A2, C2, D2, E2, G2, H2>.

3. Determining the number of vehicle used for each type

ρ = <A2, C2, D2, E2, G2, H2> uses 6 trips (A2, C2, D2, E2, G2 and H2) to perform the distribution. All these trips are using the type 2 vehicle. So, we can calculate the UB_1=0, LB_1=0 and UB_2 = 6, $LB_2 = \left\lceil \frac{10+8+8+8+10+8}{31} \right\rceil = 2$.

4. Generating solution with no-wait schedule

With the trail ρ = <A2, C2, D2, E2, G2, H2>, all trip information in Table 3.4, estimated vehicle number LB_2=2, the no-wait schedule can be built. Figure 3.20 displays the no-wait schedule for the Example 3.3 for production permutation σ = <6, 3, 2, 4, 5, 1> using H1 Algorithm.

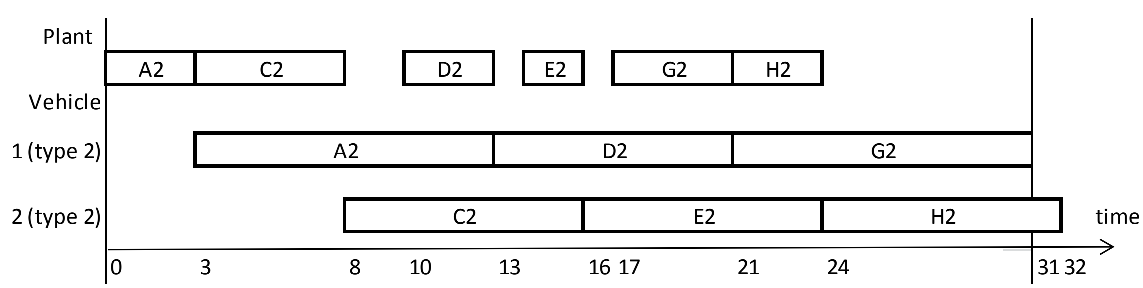

Figure 3.20: No-Wait Schedule for the Example 3.3 for Production Permutation σ = <6, 3, 2, 4, 5, 1> Using H1 Algorithm

5. Compressing the no-wait schedule

As shown in the Figure 3.20, there are some gaps in the production schedule (i.e. the gap between trip C2 and D2, gap between D2 and E2, gap between E2 and G2). The compress process is to refine the no-wait schedule to get the compressed schedule. Table 3.7 provides all trips maximum wait time which is the product lifetime minus trip delivery time. It used to measure how long the trip can keep in the temporary inventory before delivery without violating the lifetime constraint. By following the steps mentioned in previous. The compressed schedule for Figure 3.20 is displayed in Figure 3.21:

Table 3.7: All Trips Maximum Wait Time for the Example 3.3 with Production Permutation σ = <6, 3, 2, 4, 5, 1> Using H1 Algorithm

Trip name	Route	Vehicle Type	Load	Delivery Time	Travel Time	trip maximum wait time
A2	0→6→0	2	3	5	10	4
C2	0→3→0	2	5	4	8	5
D2	0→2→0	2	3	4	8	5
E2	0→4→0	2	2	4	8	5
G2	0→5→0	2	4	5	10	4
H2	0→1→0	2	3	4	8	5

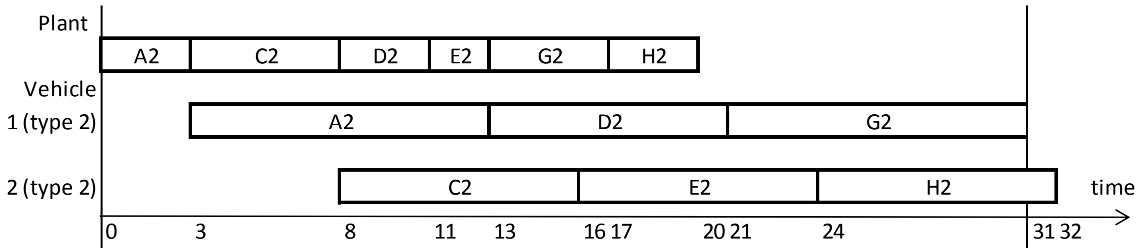

Figure 3.21: Compressed Schedule for Figure 3.20

6. Checking the feasibility of the solution

It is obvious that the solution shown in Figure 3.21 is not feasible since vehicle 2 is not meeting the time horizon constraint. Its last trip delivery will finish after the time horizon.

7. Increasing the number of used vehicles to a solution

By increasing the lower bound for type 2 vehicle, it will get the new estimate used vehicle number for type 2 vehicle LB_2=3. Repeat the previous processing, another solution can be found. The Figure 3.22 display the solution with LB_2=3.

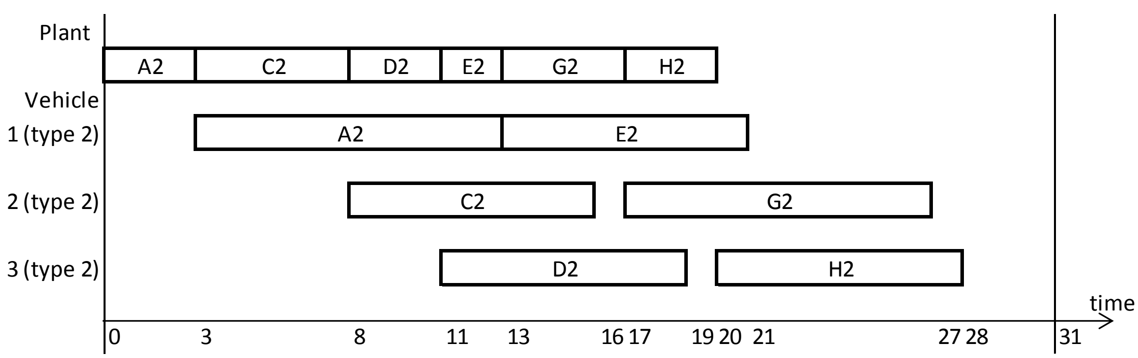

Figure 3.22: Solution for the Example 3.2 with Production Permutation σ = <6, 3, 2, 4, 5, 1> Using H1 Algorithm, LB_2=3

The solution show in Figure 3.22 is a feasible solution which implies heuristic H1 finds a feasible solution for the Example 3.2 with σ = <6, 3, 2, 4, 5, 1>. The total system transportation cost for the solution is:

Total system transportation cost = Total fixed cost + Total variable cost

= Type 1 vehicle fixed cost + Type 2 vehicle fixed cost + variable cost for trip A2

+ variable cost for trip C2 + variable cost for trip D2 + variable cost for trip E2

+ variable cost for trip G2 + + variable cost for trip H2

= 0*0 + 3*100 + 10*0.7 + 8*0.7 + 8*0.7 + 8*0.7 + 10*0.7 + 8*0.7

= 336.4

Directly Delivery with Random Fit (H2)

As shown previously, the H1 heuristic can find a unique trail for a given production permutation upon which a solution result can be built. The just fit vehicle rule, however, can lead to an imbalance in the utilization of the vehicles which makes solutions inefficient. Figure 3.23 provides an illustration. As shown in the figure, the customer delivered by trip B has a larger demand as evidenced by the fact that the plant takes longer to produce the product to satisfy it and its delivery requires a type 2 vehicle. All other trips are delivered by type 1 vehicles because the just fit vehicle rule is used to assign vehicles to routes. This creates a situation where vehicles 2 and 3 are both badly underutilized. Notice, however, if a type 2 vehicle was to be used for trip D as shown in Figure 3.24, only 2 vehicles would be needed for the trail. This improves the solution because even through the variable cost associated with delivering trip D is increased by using a larger capacity vehicle, the overall cost is less because fewer vehicles are required and the fixed costs associated with these vehicles is saved.

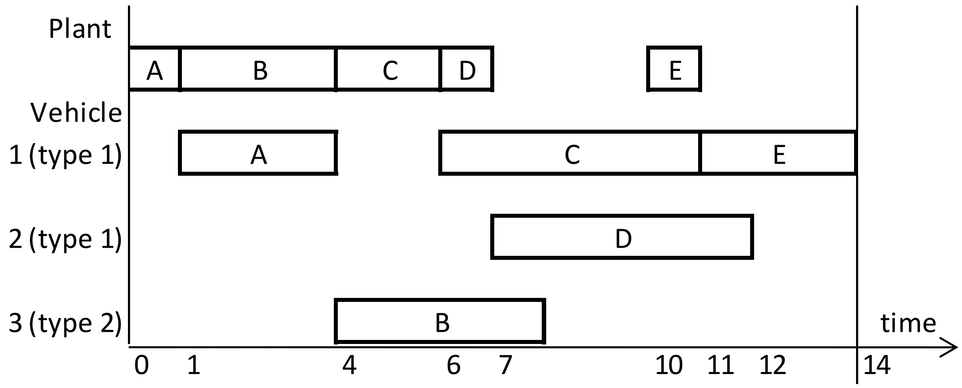

Figure 3.23: A Feasible Solution Provided by H1 Heuristic

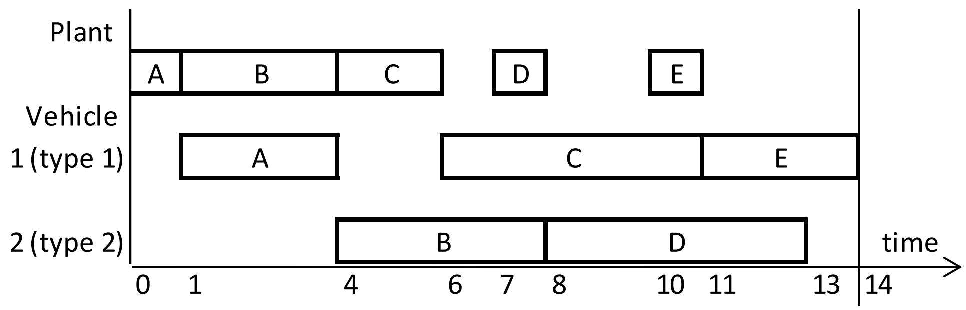

Figure 3.24: An Improved Solution of Figure 3.23

To explore the possibility of improved solutions like the one shown in the Figure 3.24, algorithm H2 is proposed that uses a random fit rule to generate the trail. The random fit rule uses a random number to determine the type of vehicle to assign when there is more than one type qualified for directly delivery. For example, Table 3.5 in Example 3.3 shows that the direct delivery trips A1 and A2 use the same delivery route but different vehicle types. The random fit rule will randomly assign the vehicle type rather than the smallest capacity vehicle possible. The remainder function is used to map the random number to a feasible trip. The function *Module*(a, b) returns the remainder of a divided by b. Using the formula

i = *Module*(*random number*, qualified vehicle type number) + 1,

the i^{th} qualified vehicle type is assigned to do the trip. For example, consider a route 0→5→0 that can be delivered by 3 different types of vehicles, {2, 3, 4}. If the random number generated is 10, then i = *Module*(10,3) + 1 = 2 or the 2nd type of qualified vehicle is selected which is type 3. The formula guarantees to map any positive integer number to a feasible vehicle type.

Multi-Stop Delivery with Shortest Path Fit (H3)

The H1 and H2 algorithms are both based on the directly delivery rule. In practice, the daily fixed cost for vehicles is typically quite high compared with variable cost so it is preferable for vehicles to deliver more than one customer demand in one trip, if possible, because it reduces the total system transportation cost. The H3 algorithm leverages this observation and generates routes based on the multi-stop delivery rule.

1. Generating a feasible trip set

    a. All feasible trips for type s vehicle

Without the delivery rule such as direct delivery, there will be more feasible trips that can be generate from a production permutation. To save all possible feasible trips generated from a production permutation σ with a type s vehicle and only considering the lifetime constraint and capacity constraint, this research defined the feasible trip graph $G_{\sigma,s}$.

Definition 3.1: Given a production permutation σ, $\sigma(i)$ is the i^{th} customer in the production permutation, where i=1,…,n, a certain type of vehicle s, $s \in S$ and a set of vertices $V(G_{\sigma,s}) = \{0, \sigma(1), ..., \sigma(n)\}$, where 0 represents the plant. $\forall \sigma(i) \in V(G_{\sigma,s})$, if the total demand of customers in the trip $0 \to \sigma(i) \to \sigma(i+1) ... \sigma(j) \to 0$ is less than the truck capacity C_s, (i.e. $\sum_{k=i}^{j} q_{\sigma(k)} \le C_s$), and all the demand on this trip could be delivered before product lifetime expires, (i.e. $\tau_{0,\sigma(i)} + \sum_{k=i}^{j-1} \tau_{\sigma(k),\sigma(k+1)} \le B$, where $1 \le i \le j \le n$), the arc $\sigma(i-1) \to \sigma(j)$ with weight $W^{s}_{\sigma(i-1),\sigma(j)} = R_s \cdot (\tau_{0,\sigma(i)} + \sum_{k=i}^{j-1} \tau_{\sigma(k),\sigma(k+1)} + \tau_{\sigma(j),0})$ will be added into the feasible trip graph $G_{\sigma,s}$, where $1 \le i \le j \le n$.

Table 3.8 enumerates all feasible trips for permutation σ = <6, 3, 2, 4, 5, 1> in Example 3.3. Figures 3.25 and 3.26 provide the feasible trips graphs for vehicle type 1 and vehicle type 2. Comparing Figure 3.25 and Figure 3.26, the former has an additional arc, B1. This is because the total demand of customers 6 and 3 (q_6+q_3=8) exceeds the capacity of vehicle 2 (C_2=7).

Table 3.8: Feasible Trip Set for Production Permutation σ = <6, 3, 2, 4, 5, 1> for Example 3.1

Trip name	Graph	Arc	Route	Weight	Vehicle Type	Load	Delivery Time
A1	$G_{\sigma,1}$	$0\to6$	$0\to6\to0$	10*1	1	3	5
B1	$G_{\sigma,1}$	$0\to3$	$0\to6\to3\to0$	12*1	1	8	8
C1	$G_{\sigma,1}$	$6\to3$	$0\to3\to0$	8*1	1	5	4
D1	$G_{\sigma,1}$	$3\to2$	$0\to2\to0$	8*1	1	3	4
E1	$G_{\sigma,1}$	$2\to4$	$0\to4\to0$	8*1	1	2	4
F1	$G_{\sigma,1}$	$2\to5$	$0\to4\to5\to0$	12*1	1	6	7
G1	$G_{\sigma,1}$	$4\to6$	$0\to5\to0$	10*1	1	4	5
H1	$G_{\sigma,1}$	$5\to1$	$0\to1\to0$	8*1	1	3	4
A2	$G_{\sigma,2}$	$0\to6$	$0\to6\to0$	10*0.7	2	3	5
C2	$G_{\sigma,2}$	$6\to3$	$0\to3\to0$	8*0.7	2	5	4
D2	$G_{\sigma,2}$	$3\to2$	$0\to2\to0$	8*0.7	2	3	4
E2	$G_{\sigma,2}$	$2\to4$	$0\to4\to0$	8*0.7	2	2	4
F2	$G_{\sigma,2}$	$2\to5$	$0\to4\to5\to0$	12*0.7	2	6	7
G2	$G_{\sigma,2}$	$4\to5$	$0\to5\to0$	10*0.7	2	4	5
H2	$G_{\sigma,2}$	$5\to1$	$0\to1\to0$	8*0.7	2	3	4

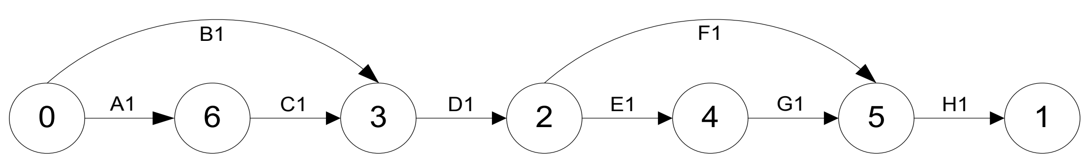

Figure 3.25: Feasible Trips Graph $G_{\sigma,1}$ for Type 1 Vehicle with Production Permutation σ = <6, 3, 2, 4, 5, 1> for Example 3.3

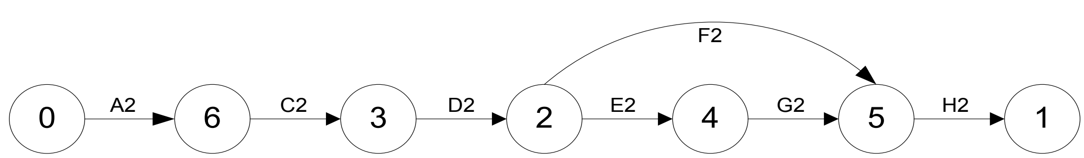

Figure 3.26: Feasible Trips Graph $G_{\sigma,2}$ for Type 2 Vehicle with Production Permutation σ = <6, 3, 2, 4, 5, 1> for Example 3.3

b. All feasible trips for all types of vehicles

Each arc in the graph $G_{\sigma,s}$ represents a feasible trip and the weight of the arc represents the total variable cost (e.g. fuel cost) for that trip. By using the same production permutation, a total of *s* graphs could be created. Each type of vehicle will generate one feasible trips graph $G_{\sigma,s}$, where $s \in S$. Define G_{σ} as a combination of graphs that combines all graphs $G_{\sigma,s}$, where $s \in S$. The vertices $<\sigma(i), s>$ in the G_{σ}, represent the i^{th} customer in the permutation which delivered by the s^{th} type vehicle. The vertices $<0,s>$ represent the plant, where $s \in S$. Also, for convenience, two dummy points: start point (*SP*) and end point (*EP*) are added. So, it has the, $V(G_{\sigma})$ = {SP, EP,…,$<0,s>$,$<\sigma(1),s>$, …,$<\sigma(n),s>$,…}, where $s \in S$. Also, set $\sigma(0)=0$. The graph G_{σ} is defined as follows:

Definition 3.2: Given a permutation σ, a group of graphs $G_{\sigma,s}$, where $s \in S$, and a set of vertices $V(G_{\sigma})=\{SP,EP,\ldots,<\sigma(0),s>,<\sigma(1),s>,\ldots,<\sigma(n),s>,\ldots\}$, where $\sigma(0)=0$ represents the plant. $\forall s \in S$, if the arc $\sigma(i-1) \rightarrow \sigma(j)$ with weight $W^{s}_{\sigma(i-1),\sigma(j)}$ is exist in the graph $G_{\sigma,s}$, where $1 \le i \le j \le n$, the arc $<\sigma(i-1), s> \rightarrow <\sigma(j), s>$ will be added into the graph G_{σ} with the weight $W^{s}_{\sigma(i-1),\sigma(j)}$. Additional dummy arcs $<\sigma(i), s> \rightarrow <\sigma(i), s+1>$, $<\sigma(i), s+1> \rightarrow <\sigma(i), s>$, where $1 \le i \le n$, $1 \le s \le |S|-1$, and $SP \rightarrow <\sigma(0), s>$, $<\sigma(n), s> \rightarrow EP$ where $s \in S$ will be added to the graph G_{σ} with weight 0.

Figure 3.27 provides the feasible trips graph G_{σ} with permutation σ = <6, 3, 2, 4, 5, 1> in Example 3.3. As shown in the figure, the arcs drawn by solid lines represent the feasible trips and the arcs drawn by broken lines represent the dummy arcs which are used to connect the graph. They carry value 0.

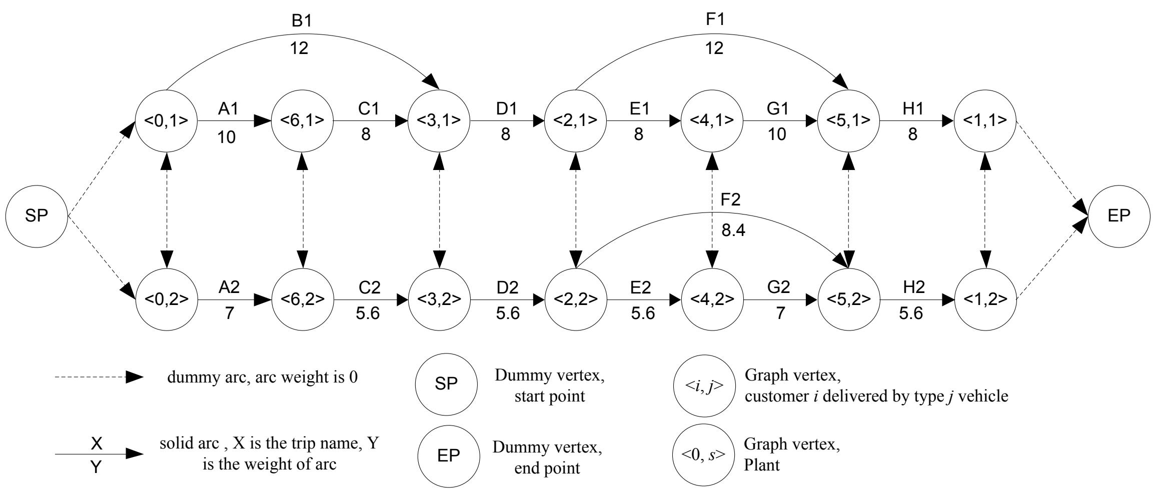

Figure 3.27: Feasible Trips Graph G_σ with Production Permutation σ = <6, 3, 2, 4, 5, 1> for Example 3.3

2. Generating a trail

In the graph G_σ, every solid arc represents a feasible trip. The path from the node *SP* to the node *EP* could generate a feasible trail to cover all customers. The weight of the path is the total system transportation variable cost which is calculated by summing all arcs weights in the path. So, the shortest path from node *SP* to the node *EP* will define the trail which covers all customers with the minimum total system transportation variable cost. By using the Dijkstra's algorithm, it is easy to find the shortest path in the graph G_σ. For example, Figure 3.28 uses the thick line drawn as the shortest path in the Figure 3.27, which is *SP*→<0,1>→<3,1>→<3,2>→<2,2>→<5,2>→<1,2>→*EP*. This path generates a trail ρ=<B1, D2, F2, H2>, and the total weight of the path is = 12*1 + 7*0.8 + 12*0.7 + 8*0.7 = 31.6.

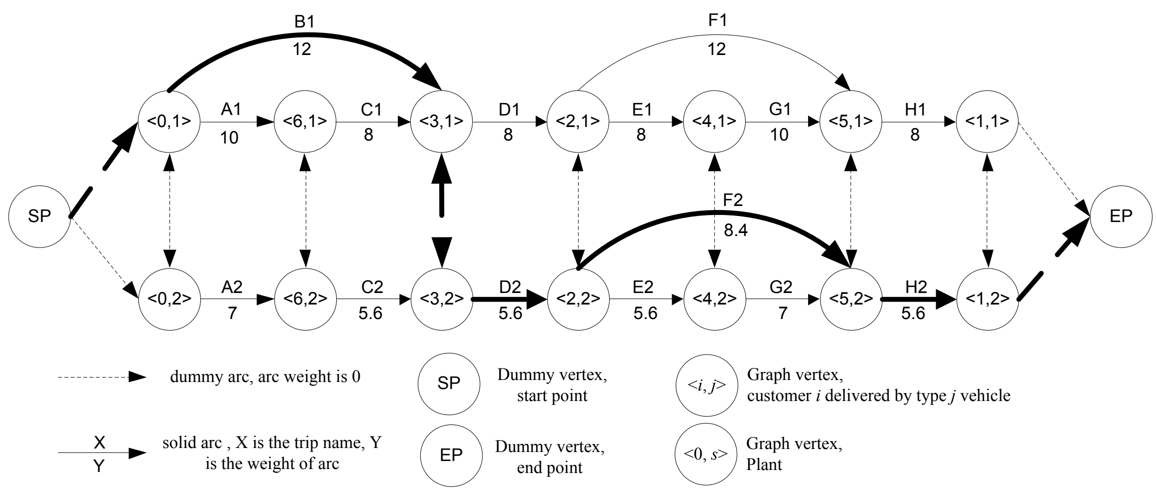

Figure 3.28: The Shortest Path from SP to EP in Figure 3.27

3. Determining the number of vehicle used for each type

With the trail ρ=<B1, D2, F2, H2> and all trips information in Table 3.8, the lower bound for each type of vehicle can be calculated. UB_1=1 and $LB_1 = \left\lceil \frac{12}{30} \right\rceil = 1$.

UB_2=3 and $LB_2 = \left\lceil \frac{8+12+8}{30} \right\rceil = 1$.

4. Generating solution with no-wait schedule

With the trail ρ = <B1, D2, F2, H2>, all trip information in Table 3.8, estimated vehicle number LB_1=1, LB_2=1, the no-wait schedule can be built. Table 3.29 shows the no-wait schedule for the Example 3.3 for production permutation σ = <6, 3, 2, 4, 5, 1> using H3 Algorithm.

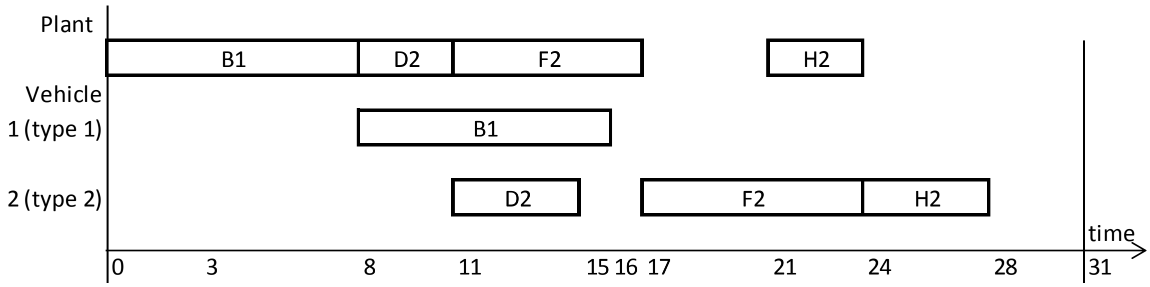

Figure 3.29: No-Wait Schedule for the Example 3.3 with Production Permutation σ = <6, 3, 2, 4, 5, 1> Using H3 Algorithm

5. Compress the no-wait schedule to last solution in iteration

As shown in the Figure 3.29, there are gaps in the production schedule (i.e. the gap between F2 and H2). The compress process is used to refine the no-wait schedule. The lifetime B=9 in Example 3.3. The refined integrated production and distribution schedule for Figure 3.30 is shown in Figure 3.29:

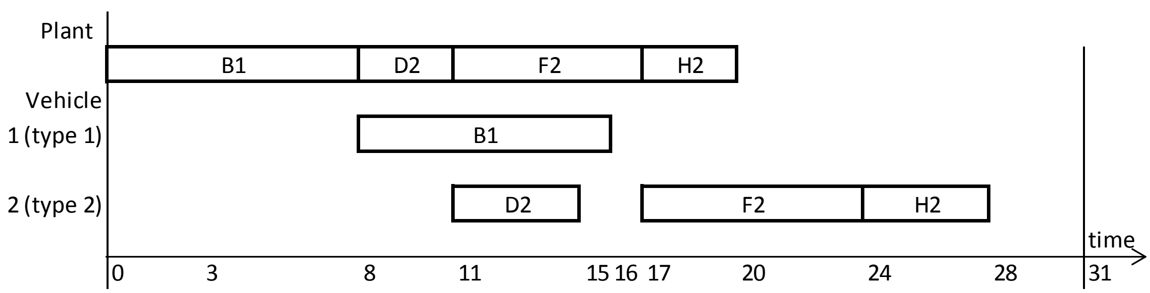

Figure 3.30: Compressed Schedule for Figure 3.29

6. Check the feasibility of the solution

The solution shown in Figure 3.30 is a feasible solution for the Example 3.3 with σ = <6, 3, 2, 4, 5, 1>. The total system transportation cost for the solution is:

Total system transportation cost = Total fixed cost + Total variable cost

= Type 1 vehicle fixed cost + Type 2 vehicle fixed cost + variable cost for trip B1 + variable cost for trip D2 + variable cost for trip F2 + variable cost for trip H2

= 1*120 + 1*100 + 12*1 + 8*0.7 + 12*1 + 8*0.7

= 255.2

Compared with the previous solution obtained by the H1 algorithm, the H3 algorithm gets a feasible solution with much less transportation cost from the same production permutation σ = <6, 3, 2, 4, 5, 1> for Example 3.3.

Multi-stop Delivery with Random Shortest Path Fit (H4) and Multi-stop Delivery with Random Fit (H5)

In the H3 algorithm, the shortest path trail provides the minimum total system transportation variable cost which is a very important factor that reduces cost in the solution. On the other hand, there are still some observations that suggest that cost of the trail can be reduced further. Like the weakness of the H1 algorithm, the trail generated by the shortest path might cause lower utilization of the last vehicle for each type. That is, a trail could increase the variable cost by using a vehicle of greater capacity that could reduce the number of vehicles required to satisfy the distribution and avoid the fixed cost of the extra vehicles. The shortest path method can lead to a good solution but improvements can be made.

Table 3.9 lists three possible trails generated by three paths. By sum all trips' weight in each path, the length of each path can be calculated. The shortest one among these three is the path $SE \rightarrow \sigma(2) \rightarrow \sigma(3) \rightarrow \sigma(4) \rightarrow EP$ and is represented by trail 3. It requires one trip with the type 1 vehicle and two trips with type 2 vehicle. When the time horizon is less than 7, i.e. $H<7$, the trip H and trip I are impossible to be assigned to one vehicle (violate the time horizon constraint) and the trail 3 needs 3 vehicles to make the delivery while the trail 2 might complete the delivery with only 2 vehicles (1 type 1 vehicle and 1 type 2 vehicle).

Table 3.9: Example of Three Paths in a Feasible Trips Graph, n=4

Trail No.	Paths from *SP* to *EP*	*Trip* name	Generated trip routes	Weight	Vehicle type
1	SP→ $\sigma(1)$ → $\sigma(2)$ → $\sigma(3)$ → $\sigma(4)$→EP	A	0→ $\sigma(1)$ → 0	2.8	1
		B	0→ $\sigma(2)$ → 0	2.8	1
		C	0→ $\sigma(3)$ → 0	2	2
		D	0→ $\sigma(4)$ → 0	1.4	1
2	SP→ $\sigma(1)$ → $\sigma(3)$ → $\sigma(4)$→EP	E	0→ $\sigma(1)$ → 0	2.8	1
		F	0→ $\sigma(2)$ → $\sigma(3)$ → 0	6	2
		G	0→ $\sigma(4)$ → 0	1.4	1
3	SP→ $\sigma(2)$ → $\sigma(3)$ → $\sigma(4)$→EP	H	0→ $\sigma(1)$ → $\sigma(2)$ → 0	5	2
		I	0→ $\sigma(3)$ → 0	2	2
		J	0→ $\sigma(4)$ → 0	1.4	1

To explore possibilities such as this, two algorithms are proposed that use randomness in the path generation process. These methods are: 1) random path heuristic (H5), and 2) two random shortest path heuristic (H4).

The H4 and H5 algorithms are similar to the H3 algorithm except they use randomness when generating the paths. The H5 algorithm randomly generates n paths to create trails and pick the one with the best objective value to be the result. By the randomness property, the random trail is able to avoid the limitation of the shortest path trail and retain the possibility of finding an improved solution such as the one shown in Table 3.9. But the H5 algorithm also has its weakness. Since an undirected complete graph has n! random paths, the change for H5 find the best solution for current graph is not large (*probability* =1/(n-1)!) and the quality of the solution is unknown (compared to the solution get from H3 algorithm). Also, H5 algorithm needs much longer solving time than the H3 algorithm.

The H4 algorithm combines some of the advantages of both the H3 and H5 algorithms. It randomly locates an *inner point* (*IP*) in the feasible trips graph and combines two shortest paths: the path from *SP* to *IP* and the path from *IP* to *EP* to be the final path to generate a trail. With this design, the H4 will spend less computing time than the H5 algorithm and retain the possibility of getting better solutions than the H3 algorithm.

Lower Bound Approach for the IPDSPPP

The lower bound discussed here is the lower bound of the minimization problem. Any feasible solution for the test problem will provide the objective value greater or equal to the LB value, no exception is allowed. So, this implies the best lower bound value for the test problem is the optimal solution. The lower bound approach is the method to calculate the lower bound value for a given problem (or a group of problems) by analysis of the basic information given by the test problem, such as customer location, demands, vehicle fixed cost and variable cost, etc. There are two important things for the LB approach: 1) provide the feasible LB, and 2) provide the LB value close to optimal solution. First, the LB approach needs to guarantee that the LB it generated for each problem is feasible. Here the feasible means that the LB generated by the LB approach is less or equal to all feasible solutions. Second, the quality of the LB is measured by the difference between LB and optimal solution. The best LB is equal to the optimal solution. So, the better LB approach can provide the LB closer to the optimal solution. In our research, the objective is to minimize the total system cost. All LB should be smaller than the optimal solution, and the LB approach which provides higher LB will be the better LB approach.

Devipriya et al. (2008) provided a lower bound approach on the objective function for IPDSPPP problem when the vehicles are identical. Estimations are made regarding the total number of vehicles required and travel distances within each trip to calculate the lower bound for the total cost of transportation. Estimating the number of vehicles allows for multiple trips and estimating the distance in each trip includes the distance between customers plus double the distance from the plant to the customer that is farthest from the plant. Here, an improved lower bound approach is proposed by considering not only the distance from the plant to customer but also the travel distance between different customers within the trip.

Theorem 1 : Consider an IPDSPPP problem with the follows attributes which:

- Perishable product with fixed lifetime *B*
- *n* customers with demands $q_i, i = 1,...,n$ and with known locations in a two dimensional plane
- Known travel times between locations, $\tau_{i,j}, 0 \le i, j \le n$, where 0 represents the plant

- production rate of r
- s different types of vehicles, $S = \{1, 2, ..., s\}$ differentiated by capacity, variable cost and fixed cost $C_i, R_i, F_i, i \in S$, respectively

Let ***G*** to be the set of $g = \left\lceil \sum_{i \in N'} q_i \Big/ \max_{i \in S} C_i \right\rceil$ customers which are closest to the plant. Then, a lower bound on the total transportation cost is:

$$LB = \min_{i \in S} R_i \cdot \left[2\sum_{i \in G} \tau_{0,i} + \sum_{i \in N'/G} \min_{k \in N'} \tau_{i,k} \right] + \min_{i \in S} F_i \cdot \left\lceil \frac{\frac{1}{r}\sum_{i=1}^{L}[(L+1-i)q_i] + 2\sum_{i \in G} \tau_{0,i} + \sum_{i \in N'/G} \min_{k \in N'} \tau_{i,k}}{H} \right\rceil$$

where $L = \left\lceil \left(2\sum_{i \in G} \tau_{0,i} + \sum_{i \in N'/G} \min_{k \in N'} \tau_{i,k} \right) \Big/ H \right\rceil$

Proof: The total transportation cost can be divided into two parts, variable and fixed costs. To address the variable cost, estimates of both the total number of trips required to deliver the product and the travel time for one trip are required. Each feasible trip covers a set of customers, $A = \{i, ..., k, ..., l\}$, $|A| \geq 1$. The route will start from the plant, visit all customers once in pre-defined route order, and then go back to the plant to end of the route. Each route can be divided into intervals defined by the inner stop points (the plant and customer locations).

Depending on the starting point for an interval, the trip intervals can be classified into 2 categories: (I) trip intervals that start or end at the plant, (II) trip intervals that both start and end at customer locations. See Figure 3.31 for an illustration. Each route must deliver product to at least one customer which means all routes contain 2 type I intervals. Hence, each route requires at least $2\min_{j\in A}\tau_{0,j}$ time units. For the rest of the customers, $A/\{j\}$, the route must have $|A|-1$ intervals to connect rest customers in the sequence. This requires at least $\sum_{i\in A/\{j\}}\min_{k\in A}\tau_{i,k}$, where $\tau_{0,j}=\min_{i\in A}\tau_{0,i}$. In general, the total travel time units of one trip is $2\tau_{0,j}+\sum_{i\in A/\{j\}}\min_{k\in A}\tau_{i,k}$. The estimated total variable cost for one trip is $R_{\min}\cdot\left(2\min_{i\in A}\tau_{0,i}+\sum_{i\in A/\{j\}}\min_{k\in A}\tau_{i,k}\right)$, where $\tau_{0,j}=\min_{i\in A}\tau_{0,i}$ and $R_{\min}=\min_{i\in S}R_i$.

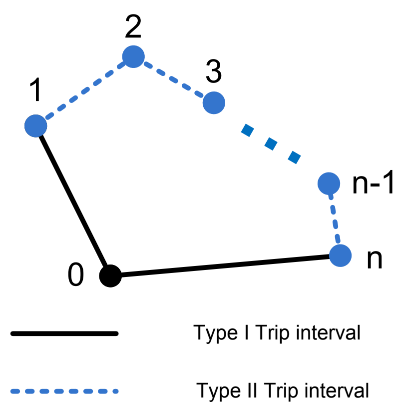

Figure 3.31: Sub-Tour Routing Structure

Now, by using the capacity restrictions on the trucks, it will take at least $g=\left\lceil\sum_{i\in N'}q_i\Big/\max_{i\in S}C_i\right\rceil$ trips to successfully deliver all products before the end of the time horizon. Let the G to be the set of g customers that are located closest to the plant. The lower bound estimate for the variable cost = $R_{\min}\cdot\left[2\sum_{i\in G}\tau_{0,i}+\sum_{i\in N'/G}\min_{k\in N}\tau_{i,k}\right]$ where

$R_{\min} = \min_{i \in S} R_i$. Note that, to ensure the feasibility of the approach, the maximal vehicle capacity is used to calculate the g value, while the minimum variable cost per mile is used to calculate the variable cost.

Determining the contribution of the fixed cost requires an estimate of the total number of vehicles used to make deliveries. Note that an initial estimate of vehicles is

$L = \left\lceil \left(2\sum_{i \in G} \tau_{0,i} + \sum_{i \in N'/G} \min_{k \in N'} \tau_{i,k} \right) \Big/ H \right\rceil$.

From Figure 3.32, it is observed that that there are gaps when trucks are idle before their first trip, d_i represent the production time for vehicle i's first trip.

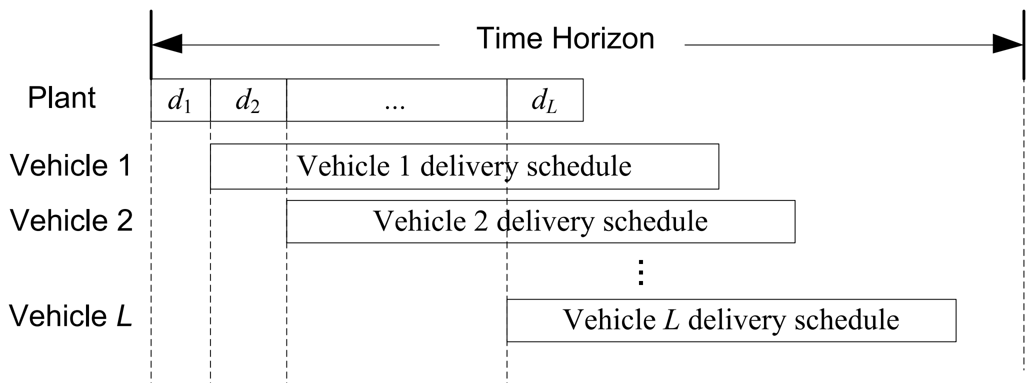

Figure 3.32: Integrated Production Schedule and Distribution Schedule Assignment

Let G' to be the set of the L smallest customer demands (q_i , $i = 1,...,L$) sorted in ascending order. Then the modified estimate is

$$L' = \left\lceil \frac{\frac{1}{r}\sum_{i=1}^{L}[(L+1-i)q_i] + 2\sum_{i \in G}\tau_{0,i} + \sum_{i \in N'/G}\min_{k \in N'}\tau_{i,k}}{H} \right\rceil$$

Finally, the total fixed setup cost = $\min_{i \in S} F_i \cdot L'$ and the lower bound for the objective function is:

$$LB = \min_{i \in S} R_i \cdot \left[2\sum_{i \in G} \tau_{0,i} + \sum_{i \in N'/G} \min_{k \in N'} \tau_{i,k} \right] + \min_{i \in S} F_i \cdot \left\lceil \frac{\frac{1}{r}\sum_{i=1}^{L}[(L+1-i)q_i] + 2\sum_{i \in G} \tau_{0,i} + \sum_{i \in N'/G} \min_{k \in N'} \tau_{i,k}}{H} \right\rceil$$

Numerical Analysis

Structure of the analysis

In the previous section, new heuristic algorithms and a lower bound were proposed. In this section, an experimental study with numerical examples is conducted to demonstrate the usefulness of the heuristics, compare performance of the heuristics as measured by the quality of the solutions they generate, and analyze the quality of the new lower bound approach method.

To analyze the performance of the new LB approach, a comparison analysis is performed that compares the LB value provide by the old approach and the new approach (A1). Since the feasibility of the LB approach had been proved, this analysis (A1) is used to check whether the new approach can provide a higher LB than the older approach. The analysis not only compares the value of the lower bound, it also compares each of the components of the lower bound: travel distance, which relates to the variable cost, and the used vehicle number which relates to the fixed cost. Since the old lower bound is restricted to identical vehicles, the comparison is made in the identical vehicle environment.

Another comparison analysis is performed between the optimal solution, heuristic solutions and LB value (A2) for problems that are sufficiently small for optimal solutions to be found. In this research, that is 7 customers or less. The fact that IPDSPPP is NP-hard justifies the use of heuristic algorithms to solve the problem but providing a measure of solution's quality is important. At the same time, the LB quality is check for those problems. In order to reflect the quality of the heuristic solution and LB, three gaps are presented: 1) optimality gap. This is the gap between a feasible solution and the optimal solution. It is used to reflect the quality of the feasible solution.

$$Optimality\ gap = \frac{feasible\ Solution - optimal\ solution}{optimal\ solution}$$

2) LB quality gap. This is the gap between the optimal solution and LB value generated by the LB approach. This gap is used to reflect the quality of the LB approach. The LB gap equal 0 means that the LB approach is very suitable for this problem and provides the best LB value (i.e. optimal solution).

$$LB\ quality\ gap = \frac{optimal\ solution - Lower\ bound}{optimal\ solution}$$

3) LB Gap: this is the gap between the feasible solution and LB value. This gap has the similar function with the optimality gap which is used to reflect the quality of the feasible solution. The optimal solution is replaced by the LB value. Note that, this LB gap might not reflect the quality of the feasible solution accurately when the LB has a large deviation from the optimal solution.

$$LB\ gap = \frac{feasible\ Solution - Lower\ bound}{Lower\ bound}$$

In the A2 analysis, we will compare the optimality gap between the optimal solution and Best Heuristic Solution which is defined as the best heuristic solution among all solutions using all heuristics. This comparison will provide the quality of the heuristic algorithms proposed in this research. Also, the LB quality will be checked by the LB quality gaps between optimal solution and LB value. In total, 15 problems are divided into 5 groups based on the number of customers in the problem (i.e., one group for each number of customers from 3 to 7). In each group, 3 problems are created with the same customer number, but random customer location and customer demand. The MIP solver with AMPL + GRUOBI is used to solve the optimal solution and 5 heuristics (H1, H2, H3, H4 and H5) are used to get the Best Heuristic Solution.

The experimental study to compare the performance of the heuristics consists of 16 test problems that are divided into 2 categories and 5 heuristics (H1, H2, H3, H4 and H5). There are 4 small size problems which include no more than 7 customers because this is the maximum number of customer for which the optimal solution to the MIP model could be found with the computing environment used in this research. There are 12 large size problems which have more than 7 customers. Since the heuristics are affected by the random number used in the heuristic, the same input environment will likely generate different results so, for each problem in this analysis, 50 replications are used with each heuristic.

To compare the quality of the solution produced by different heuristic algorithms, a comparison study is performed (A3). Different from the A2 analysis, in the A3 analysis more detailed information is compared which includes the minimum, maximum and average cost objective value considering all 50 runs for each problem. Statistical analysis is used to define which of the 5 heuristics performed best using the different measures.

The first 3 analyses used test problems that were differentiated only by the number of customer in the system. This will certainly increase the total transportation cost because more customers must be served. There are, however, other factors that might affect the solution result, such as the ratio of the fixed and variable transportation costs, the product lifetime length, and the typical customer order size. The heuristic algorithms are compared relative to these parameters. Analysis (A4) explores the impact of input parameters on the heuristic's performance. In particular, the product lifetime, the plant production rate, the fixed cost for each vehicle, the variable cost for each vehicle will be analyzed.

Experimental Results

1. A1 analysis.

This A1 analysis compares the performance of the new LB approach with the old one. Recall from the previous section, the LB is the lower limit of solution objective value (i.e., the system transportation cost in IPDSPPP). For a minimization problem, the LB value will be less than or equal to the objective function value for all feasible solutions. The LB approach is used to provide the LB with 2 characteristics: 1) guaranteed to be less or equal to all feasible solution, and 2) as big as possible. The first characteristic is usually guaranteed when developing the approach, such as the new approach proposed in this research. The second characteristic is used to compare the performance of the LB approach. The better approach will provide higher LB value. In this research, the LB value is the system transportation cost which is composed by 2 parts: fixed cost and variable cost which are determined by the used vehicle number and vehicle travel distance. The new LB approach in this research changed in two ways in an effort to improve its performances: (1) the internal travel distance between customers was included rather than simply doubling the farthest customer traveling distance to calculate the travel distance for trips. (2) The waiting time for all available vehicles at the beginning of the time horizon was included to improve the accuracy of estimation for used vehicle number. In order to investigate if these changes actually improved the quality of the new LB, a group of 80 problems has been generated by varying the number of customers (16 levels) and randomly generating their locations (5 problems for each number of customers). The basic parameters for these problems are shown in Table 3.10. For the comparison to be "fair" with the old LB, only one type of vehicle is used for delivery since the old LB was developed based on this assumption. The time horizon is automatically adjusted to maintain the feasibility of the solution with the formula $T = \lceil N/10 \rceil * 100 + 200$, where N is the customer number in the system. The results in run order are displayed in the Appendix, Table A.1.

Table 3.10: Input Parameters for A1 Analysis

Parameters	Values
Customer demand	$q_i \sim U(1, 15)$
Number of customer	N
Customer location range	(x, y), $0 \le x \le 100$, $0 \le y \le 100$
Plant location	(50, 50)
Plant production rate	r=1
Vehicle capacity	C=15

Vehicle variable cost per unit distance	R=1
Vehicle fixed cost	F=500
Time horizon	$\lceil N/10 \rceil * 100 + 200$

As described above, either criteria improvement (travel distance or used vehicle number) can make the LB value improved. To make the comparison, this analysis uses the data from the old approach as the benchmark and defines 3 test criteria: 1) LB value, 2) used vehicle number and 3) travel distance. The improvement for each criterion is defined as the difference between the data (i.e. LB value, used vehicle number and travel distance) generated using the new LB approach and the data using the old one. Figures 3.33, 3.34, 3.35 plot this data and show the improvement of three criteria across the 80 test problems. As shown in these figures, most of the test problems have positive improvement for all three criteria, and only three test problems have the negative improvement in LB value and travel distance (these are problems 8, 11, and 14 in Appendix Table A.1). This performance suggests that the new approach provides a better LB in most test problems. Also, we can see several cases have a significant improvement for the new LB in the figure 3.33. By going through figure 3.33 and 3.35 together, we can find that in all these cases with significant improvement, the new LB approach estimate the used vehicle number is larger than the old LB approach. In this research, the new LB approach had a better estimation of traveling distance and idle time for vehicle waiting time, which leads to a better estimation of number of used vehicles. So, when the number of used vehicle had a significant influence to the total transportation cost, the new LB approach will give a better performance.

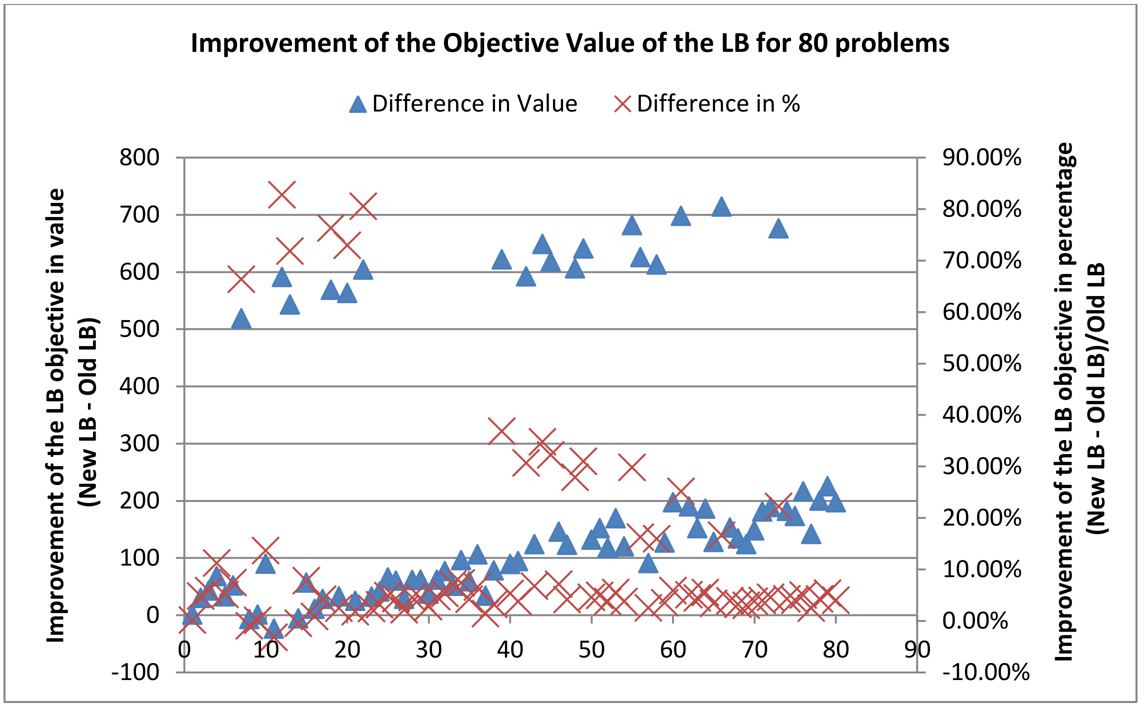

Figure 3.33: Improvement of LB Value for 80 Test Problems

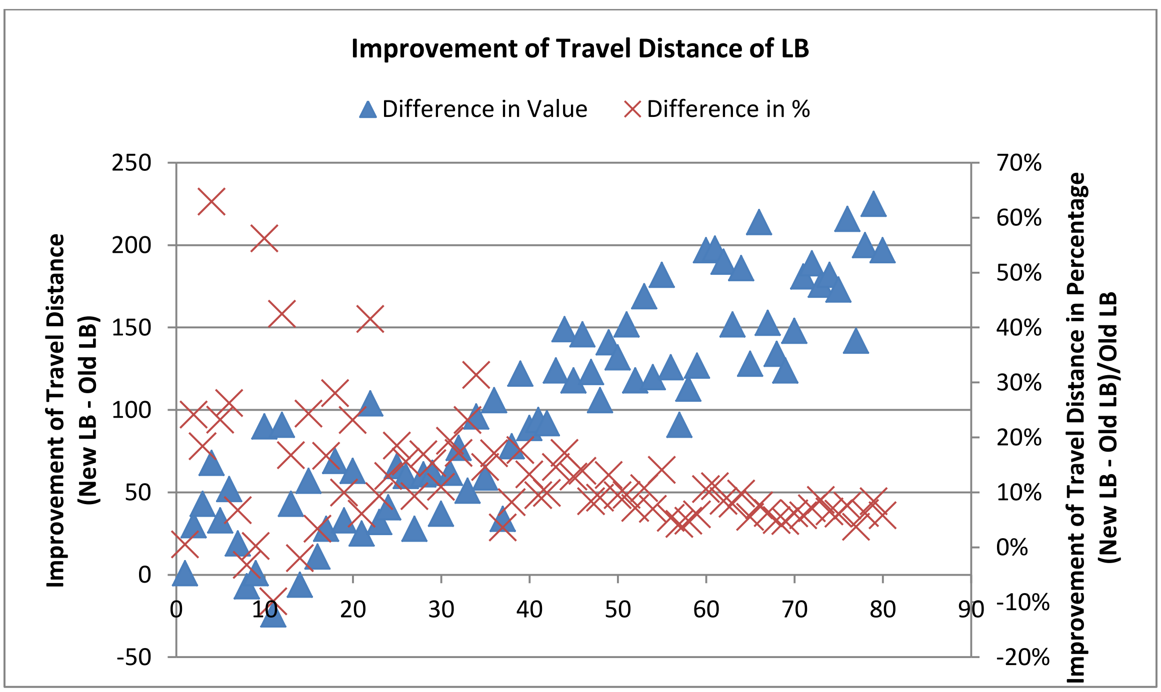

Figure 3.34: Improvement of Travel Distance for 80 Test Problems

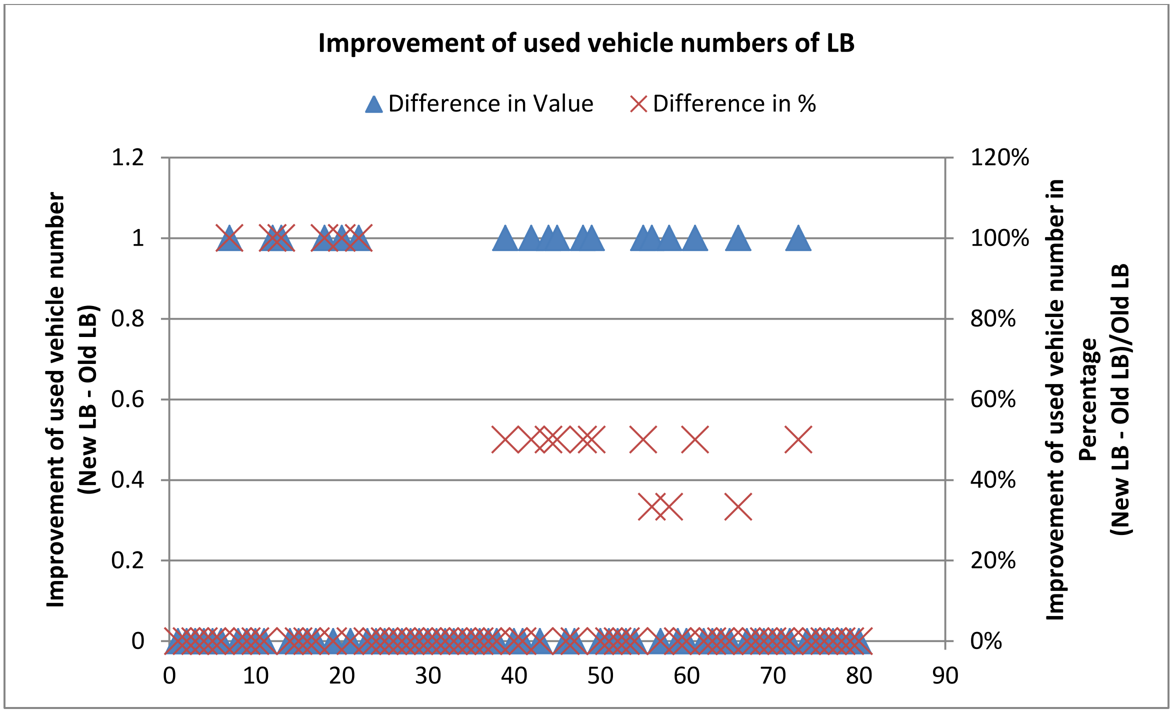

Figure 3.35: Improvement of the Used Vehicle Number for 80 Test Problems

To compare the results statistically, the raw data (in Table A.1) for the total transportation cost, the number of vehicles used, and the total travel distance was subjected to the Anderson-Darling test to check for normality. The test results show that neither the improvement of LB nor the improvement of used vehicles number are normally distributed but the improvement of travel distance is normal distributed. The probability plots for all three measures are found in Figures A.2 through A.4 found in Appendix A. Hence, a nonparametric test is required for the LB value and used vehicle number. In this research, the Wilcoxon Signed-Rank test was selected to test if the medians of LB value and used vehicle number are the same or different for the different LB approach. A paired t test is used to check if the means of travel distance given by two

LB approach are the same or not. The hypotheses tested are shown in Table 3.11. Minitab was used for testing and the results are shown in Figures A.5, A.6 and A.7. The *p*-values of all tests are less than 0.001 which means that there is sufficient evidence to reject the null hypothesis and conclude that the new approach produces better LB results in all three measures when compared to the old approach.

Table 3.11: Hypotheses Table for All Tests in A1

Improvement in LB value	H0: median of the LB value provide by new approach and old approach are equal H1: median of the LB value provide by new approach is greater than old approach
Improvement in used vehicle number	H0: median of the used vehicle number provide by new approach and old approach are equal H1: median of the used vehicle number provide by new approach is greater than old approach
Improvement in travel distance	H0: mean of the travel distance provide by new approach and old approach are equal H1: mean of the travel distance provide by new approach is greater than old approach

2. A2 Analysis

The A2 analysis is used to check the quality of the heuristic solutions and LB given by the new approach. Recall that analysis A2 compares the 5 heuristics for small problems. 3 different problems have been generated for 5 different customer sizes (i.e., number of customers is 3, 4, 5, 6, and 7). The problems for a given number of customers have locations and demands randomly generated. The rest of the input parameters settings are listed in Table 3.12. Each of these problems can be solved to optimality with MIP model by using the current software and hardware. The heuristic solutions are generated by using 5 different heuristics (H1, H2, H3, H4 and H5). The best heuristic solutions getting among the 5 heuristics are saved as Best Heuristic Solution. So the quality of the heuristic solution for this research is computed by the optimality gap:

$$optimality\ gap = \frac{Best\ Heuristic\ Solution - optimal\ solution}{optimal\ solution}$$

Also, the LB quality gap is used to compare the quality of the LB:

$$LB\ quality\ gap = \frac{optimal\ solution - Lower\ bound}{optimal\ solution}$$

In this analysis, two different types of vehicles are used for distribution and all 5 heuristic methods are applied to each problem. Each heuristic is applied to each problem 50 times and the minimum cost solution of these 50 is the defined as the best solution for that heuristic. The minimum cost solution of the minimums from the 5 heuristics is defined as the Best Heuristic Solution for that problem and used to compute the gap. That is, the Best Heuristic Solution is the best solution from the 5 heuristics * 50 applications of each = 250 total runs. The results for all runs are reported in Table 3.13.

Table 3.12: Input Parameters for A2 Analysis

Parameters	Values
Customer demand	$q_i \sim U(1, 15)$
Total Number of customer	3,4,5,6,7
Customer location range	(x, y), 0≤x≤100, 0≤y≤100
Plant location	(50, 50)
Plant production rate	r=1
Vehicle capacity	C_1=12, C_2=15
Vehicle variable cost per unit distance	R_1=1, R_2 =1.1
Vehicle fixed cost	F_1=500, F_2 =600
Time horizon	400

Table 3.13: All Running Results for A2 analysis

Problem Number	Customer Number	Optimal Solution		Best Heuristic Solution			LB	
		Solution	Time (s)	Solution	Time (s)	Optimality gaps (%)	LB	LB quality gaps (%)
1	3	680	3.9	680	0.84	0	640	6
2	3	800.2	58.4	800.2	0.73	0	656	18
3	3	911.3	5.9	911.3	0.78	0	696	24
4	4	827	12861	827	1.03	0	676	18
5	4	717	2130	717	0.89	0	702	2
6	4	799	12711	799	0.94	0	702	12
7	5	888	68904	888	1.21	0	706	20
8	5	860.7	13049	860.7	1.02	0	680	21
9	5	1010.3	1412	1010.3	1.13	0	766	24
10	6	1534.4	35028	1534.4	1.35	0	840	45
11	6	831	18675	831	1.48	0	702	16
12	6	971.8	84302	971.8	1.40	0	716	26
13	7	1652	96323	1652	1.43	0	1422	14
14	7	1579.6	73686	1579.6	1.58	0	860	46
15	7	1608.6	82835	1608.6	1.56	0	874	46

Note: the solving time is measured in seconds.

Figure 3.36, which uses the data from Table 3.13, compares the Best Heuristic Solution and LB to the optimal solution for the 15 problems. The comparison results of the optimality gaps show that at least 1 of the heuristics found the optimal solution in all cases as evidenced by a gap of 0 for each problem. This suggests that the heuristic algorithms perform well for small problems. The LB quality gaps result shows that the quality of the LB provided by the new LB approach is not stable. It provides a few good LB values, such as problem 5 only has the LB quality gap equal to 2%, while for some other problems the LB quality gaps are much larger, such as the problem 10, 14 and 15 all have the LB quality gap greater than 45%. On average, the LB quality gap for the 15 problem is equal to 23%. There are many reasons for this; one is that the LB approach does not consider the lifetime constraint which can make some solutions infeasible. Also, the approach uses the maximal capacity vehicle to calculate the used vehicle number. The minimum variable cost and fixed cost to calculate the transportation cost may also be responsible for the gaps.

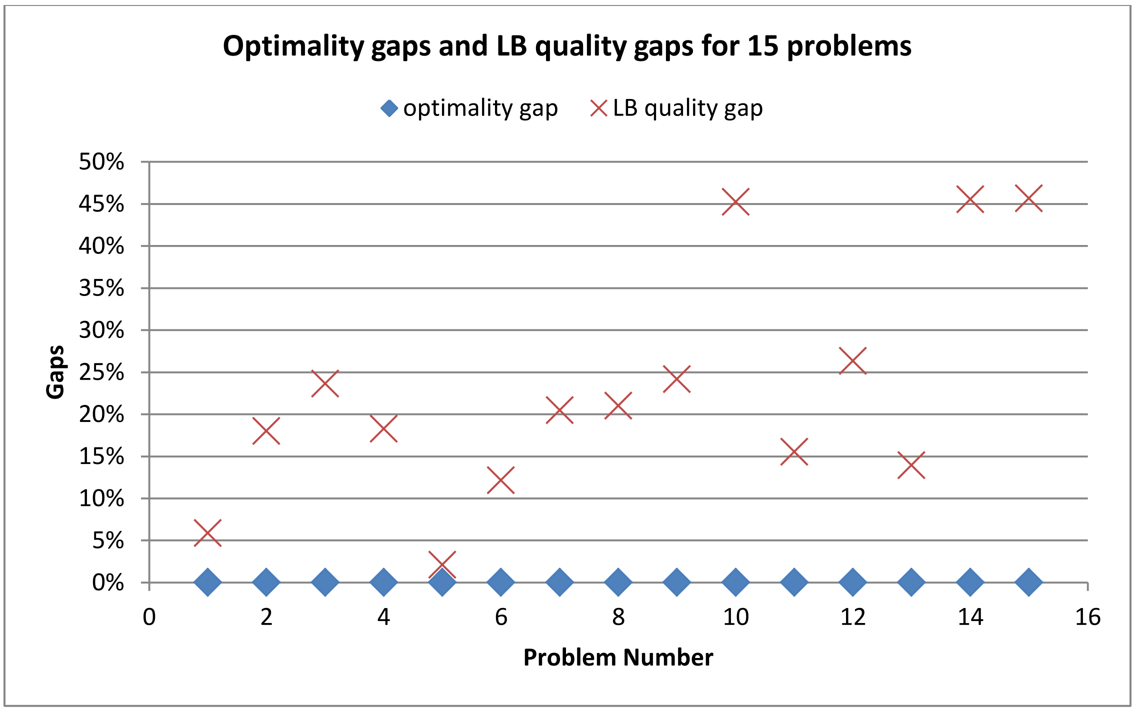

Figure 3.36: Optimality Gaps and LB Quality Gaps for 15 Test Problems in A2 Analysis

Table 3.13 contains the raw data from the analysis. Note the dramatic differences in computation time required by the heuristic algorithms and the computation time for the optimization software. Note that the solving time is not just for the single run that found the optimal solution; it is the total time to solve all of the problems; that is, the total time to resolve all 250 runs across the 5 heuristics. This total time for the heuristics is less than

2 seconds while the average solving time for the mixed integer programming model required several hours. Figure 3.37 shows the number of times each heuristic provided the Best Heuristic Solution which, in these runs, was the optimal solution. If two heuristics (or more) both provide the optimal solution, the "frequency" value in Figure 3.37 for both heuristics will be increased by 1. Notice that the H5 algorithm provided the Best Heuristic Solution every time. This is because the H5 algorithm has no extra rule while building the feasible trip set and finding the trail, it is possible to generate more different solutions than other heuristics (i.e. H1, H2, H3 and H4). So the H5 heuristic has the possibility to enumerate all solutions, and it proved that works fine when problem size is small (≤ 7 customers).

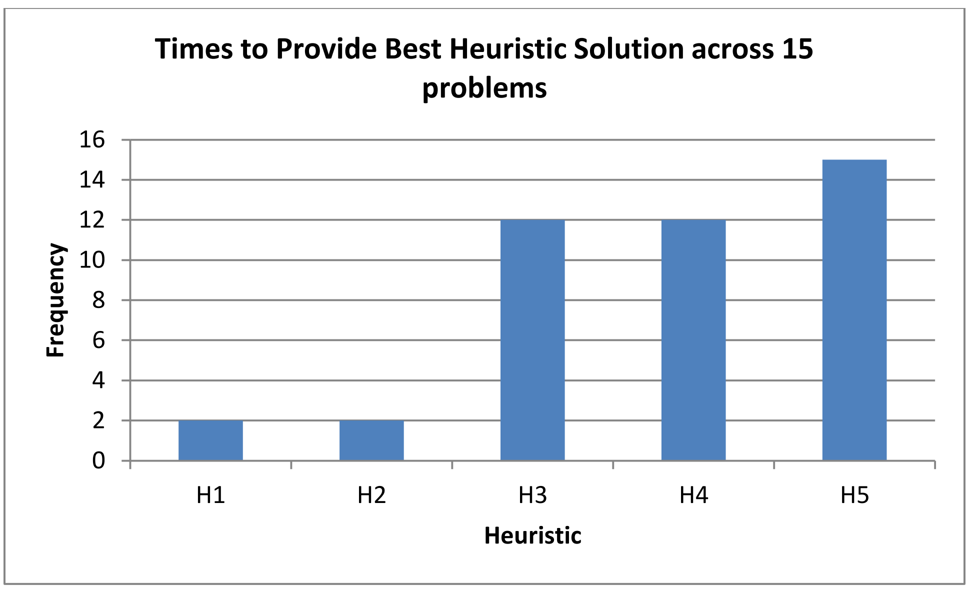

Figure 3.37: Times to Provide Best Heuristic Solution for Each Heuristic across 15 Problems

3. A3 Analysis

Analysis A3 compares the 5 heuristic algorithms against each other. Different with the MIP method, the heuristic algorithms cannot be guaranteed to find the optimal solution. At the same time, every heuristic algorithm in this research applied randomness in the process, which means that the solution of heuristics cannot always be repeated in all iterations. So, for every case, m iterations will be applied for each heuristic algorithm. Three measure criteria are achieved from these m iterations: Minimum Objective Value (reflects best case), Maximum Objective Value (reflect worst case) and Average Objective Value (reflect average case). The comparison between different heuristics is all based on these three criteria. Since the objective is to minimize the total system transportation cost, for each criterion, the heuristic which provides the minimum value will be the best heuristic. In this analysis, a total of 16 problems are considered – 4 with 7 customers or less that we henceforth refer to as small and 12 with 8 customers or more that we refer to as large. Since the heuristics contain random features, each problem-heuristic pair is replicated 50 times. The parameters used in this analysis are displayed in Table 3.14. The minimum cost, maximum cost and average cost of the 50 replications is recorded and reported in Tables 3.15, 3.16 and 3.17. The Best column saves the best result among 5 heuristic for the minimum cost, maximum cost and average cost. Also, the LB is added in each table, the quality of the Best result is reflected by LB gaps which compares between the LB and the best result.

$$LB\ gap = \frac{best\ heuristic\ result - Lower\ bound}{Lower\ bound}$$

Table 3.14: Input Parameters for A3 Analysis

Parameters	Values
Customer demand	$q_i \sim U(1, 15)$
Total Number of customer	N
Customer location range	(x, y), $0 \le x \le 100$, $0 \le y \le 100$
Plant location	(50, 50)
Plant production rate	r=1
Vehicle capacity	C_1=12, C_2=15
Vehicle variable cost per unit distance	R_1=1, R_2 =1.1
Vehicle fixed cost	F_1=500, F_2 =600
Time horizon	$\lceil N/10 \rceil * 100 + 200$

Note that the LB gaps in these 3 tables are large (94% on average). There are 2 reasons can cause this: 1) the quality of the heuristic solution is not good. 2) The LB quality is not good. Recall the LB quality analysis in A2 analysis. It shows that there are many place need to improve in the LB approach. So, later in this research, it will just use the absolute value comparison between heuristics to choose the best heuristics.

Figures 3.38, 3.39 and 3.40 report the minimum cost, maximum cost and average cost determined by the 5 different heuristics for the 16 problems in the run order found in Table B.1, B.2 and B.3. The obvious trend in all three figures is that the total cost increases as the number of customers increases, an obviously correct conclusion. To better see the performance of the different heuristics, Figures 3.41, 3.42, and 3.43 show the number of times each heuristic had the best solution of the 5 for minimum cost, maximum cost and average cost, respectively. The H3 and H4 heuristic are best most often when performance the minimum cost, average cost, the maximum cost. One the other hand, H1 and H2 does not have any best record in any performance with these three criteria. This suggests that H3 and H4 are better than H1 and H2. Attention is now turned to a statistical analysis of the data.

Table 3.15: All Running Result for A3 Analysis with Minimum Objective Value for 15 Problems

Problem Number	Minimum Value					Best	LB	LB gap
	H1	H2	H3	H4	H5			
1	764	890.4	751	751	751	751	651	15
2	1326	1438.2	758	758	758	758	650	17
3	2022	2260.4	1441	1438	1438	1438	738	95
4	2069.2	1594	1467.9	1473.9	979.5	979.5	688	42
5	2292	2299.4	1674.2	1680.3	1697	1674.2	875	91
6	1586	1724.2	1480	1462	1571.9	1462	807	81
7	2357.6	2492.4	1659.1	2272.3	1743.2	1659.1	882	88
8	3635	3791	3485	2913.9	2941.9	2913.9	1790	63
9	4999.8	5163	4023.4	3430.5	4236.1	3430.5	1956	75
10	5798	5995	4847.6	4757.1	5111.7	4757.1	2448	94
11	6536.8	6910.2	5956	5897.5	5733.5	5733.5	2520	128
12	8099	8498.2	7086.7	7059.9	7549.1	7059.9	3563	98
13	8519.6	8798.6	7412.4	7264.1	7793.4	7264.1	3766	93
14	9090.2	9473.6	7603.2	7611.2	8143.4	7603.2	3261	133
15	9685.4	10144	8012.8	7886.1	9049	7886.1	3176	148
16	11753.8	12276.8	9587.3	10161.1	11108.8	9587.3	5185	85

Table 3.16: All Running Result for A3 Analysis with Maximum Objective Value for 15 Problems

Problem Number	Maximum Objective Value					Best	LB	LB gap
	H1	H2	H3	H4	H5			
1	764	890.4	751	751	755	751	651	15%
2	1326	1438.2	758	760	1448.2	758	650	17%
3	2022	2260.4	1441	1441	2198.3	1441	738	95%
4	2069.2	1594	1467.9	1473.9	979.5	979.5	688	42%
5	2292	2305.2	1680.3	1680.3	2252	1680.3	875	92%
6	1586	1724.2	1480	1485	1724.7	1480	807	83%
7	2357.6	2492.4	2272.3	2272.3	2453.8	2272.3	882	158%
8	3635	3799.2	3485	3485	3251.6	3251.6	1790	82%
9	4999.8	5172.4	4209.1	4179.4	4610.8	4179.4	1956	114%
10	5798	6155.8	5516.2	5501.1	6015	5501.1	2448	125%
11	6536.8	6921.2	6140.9	6132.6	6585.2	6132.6	2520	143%
12	8099	8657.2	7737.2	7748.9	8509.1	7737.2	3563	117%
13	8519.6	8948.2	8078.5	7645.6	8579.4	7645.6	3766	103%
14	9090.2	9973.6	8290.6	8585.7	9428.7	8290.6	3261	154%
15	9685.4	10357	8805.9	9020.4	9868.7	8805.9	3176	177%
16	11753.8	12293.8	10961	11105.2	12095.3	10961	5185	111%

Table 3.17: All Running Result for A3 Analysis with Average Objective Value for 15 Problems

Problem Number	Average Objective Value					Best	LB	LB gap
	H1	H2	H3	H4	H5			
1	764.0	890.4	751.0	751.0	754.9	751.0	651	15%
2	1326.0	1438.2	758.0	758.0	1434.4	758.0	650	17%
3	2022.0	2260.4	1441.0	1440.3	2168.8	1440.3	738	95%
4	2069.2	1594.0	1467.9	1473.9	979.5	979.5	688	42%
5	2292.0	2300.7	1680.1	1680.3	2240.9	1680.1	875	92%
6	1586.0	1724.2	1480.0	1475.8	1721.6	1475.8	807	83%
7	2357.6	2492.4	2260.0	2272.3	2439.6	2260.0	882	156%
8	3635.0	3796.9	3485.0	3029.4	3204.8	3029.4	1790	69%
9	4999.8	5165.8	4127.4	3891.6	4592.6	3891.6	1956	99%
10	5798.0	6104.3	5298.3	5010.4	5843.9	5010.4	2448	105%
11	6536.8	6914.4	6051.6	6043.9	6485.0	6043.9	2520	140%
12	8099.0	8600.0	7543.9	7515.6	8190.7	7515.6	3563	111%
13	8519.6	8928.1	7608.8	7448.5	8258.0	7448.5	3766	98%
14	9090.2	9753.0	7843.0	8049.9	8796.3	7843.0	3261	141%
15	9685.4	10258.6	8307.3	8470.8	9545.8	8307.3	3176	162%
16	11753.8	12280.0	10327	10536.0	11604.5	10327.4	5185	99%

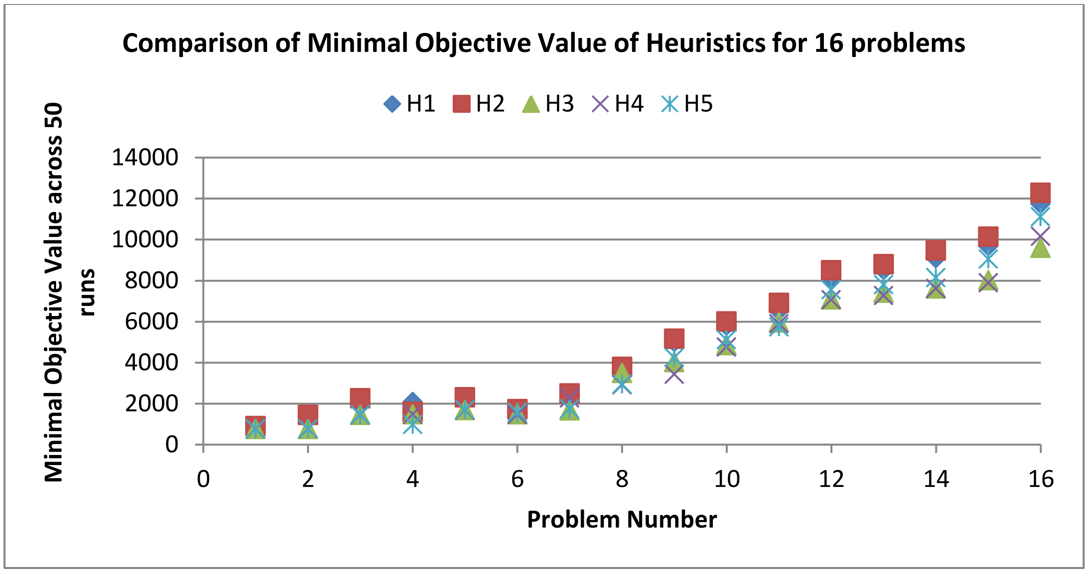

Figure 3.38: Comparison of Minimum Objective Value of Heuristics for 16 problems in A3 Analysis

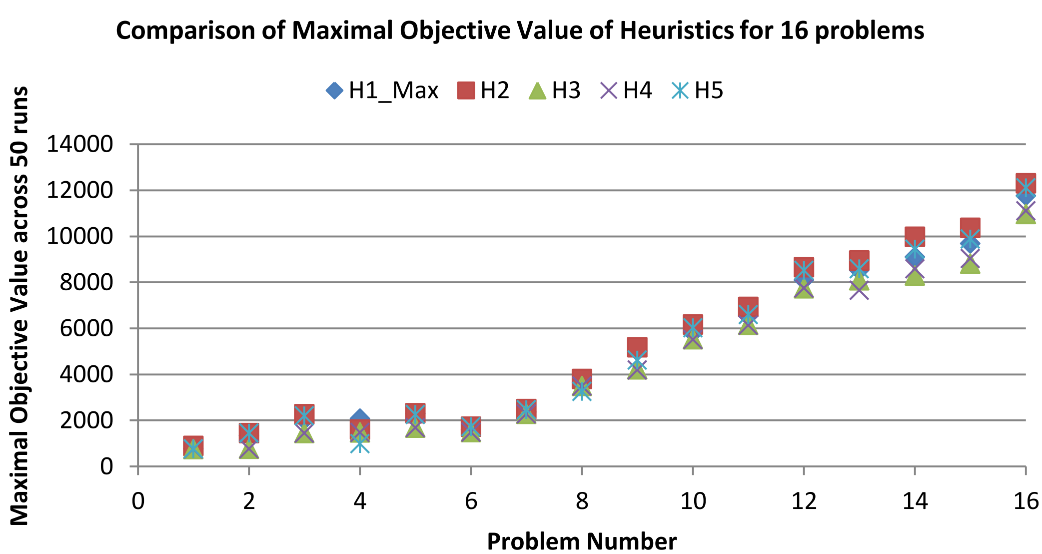

Figure 3.39: Comparison of Maximum Objective Value of Heuristics for 16 problems in A3 Analysis

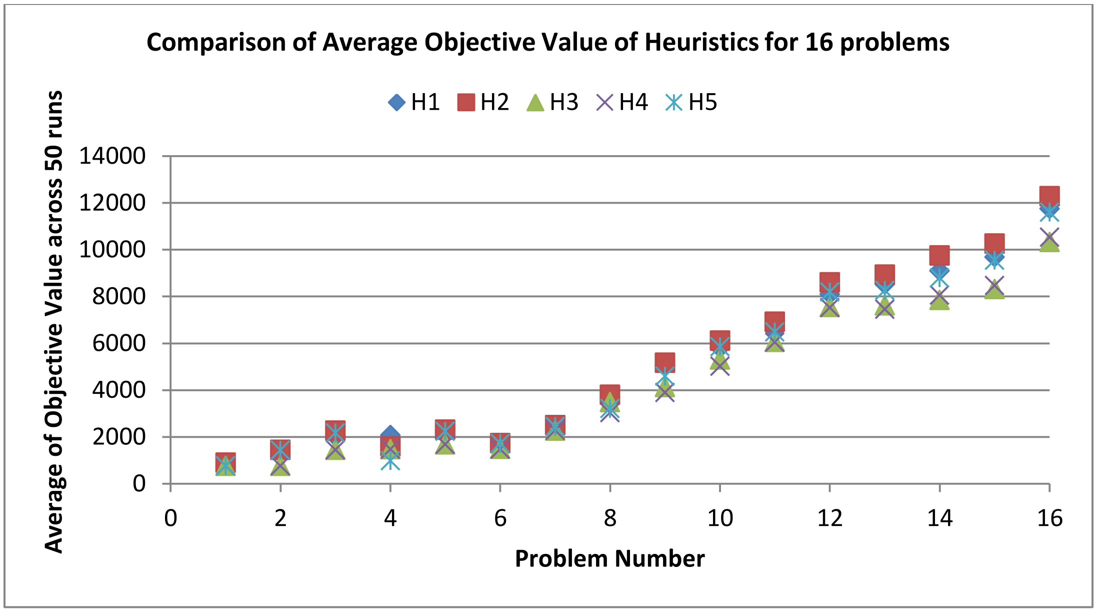

Figure 3.40: Comparison of Average Objective Value of Heuristics for 16 problems in A3 Analysis

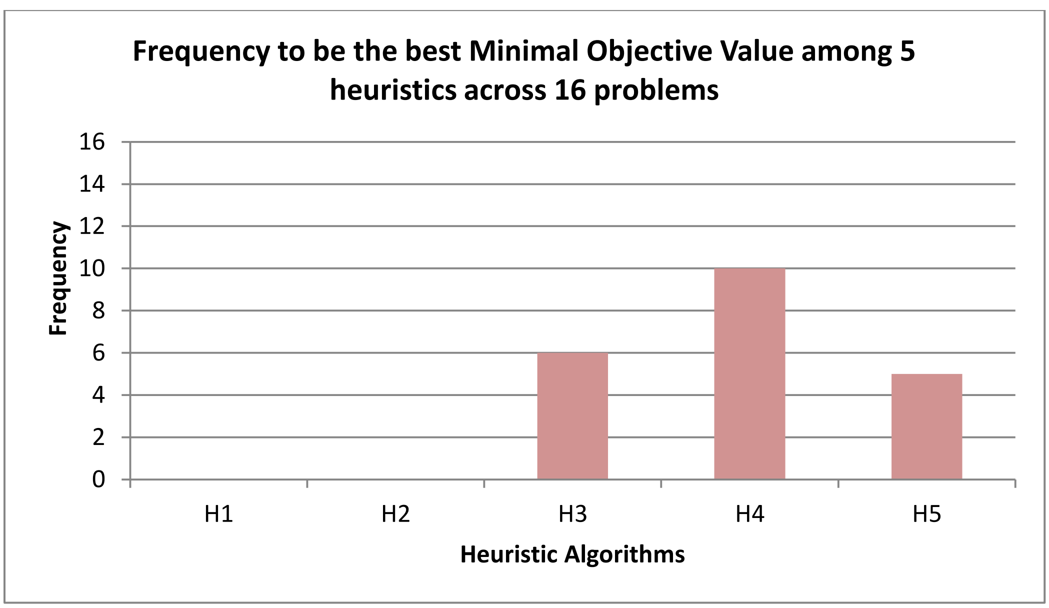

Figure 3.41: Frequency to Be the Best Minimum Objective Value among 5 Heuristics across 16 Problems

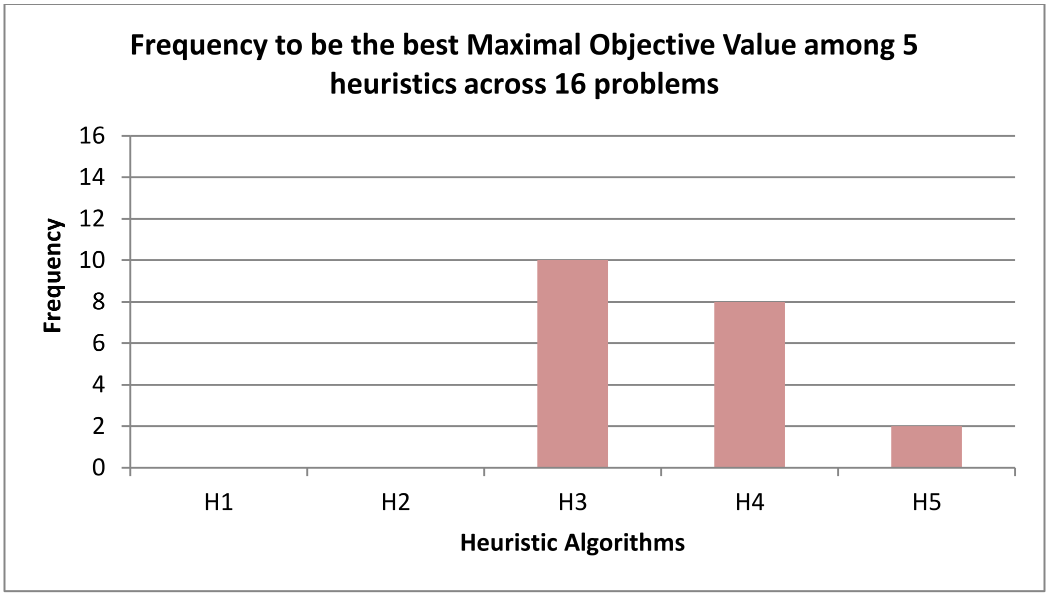

Figure 3.42: Frequency to Be the Best Maximum Objective Value among 5 Heuristics across 16 Problems

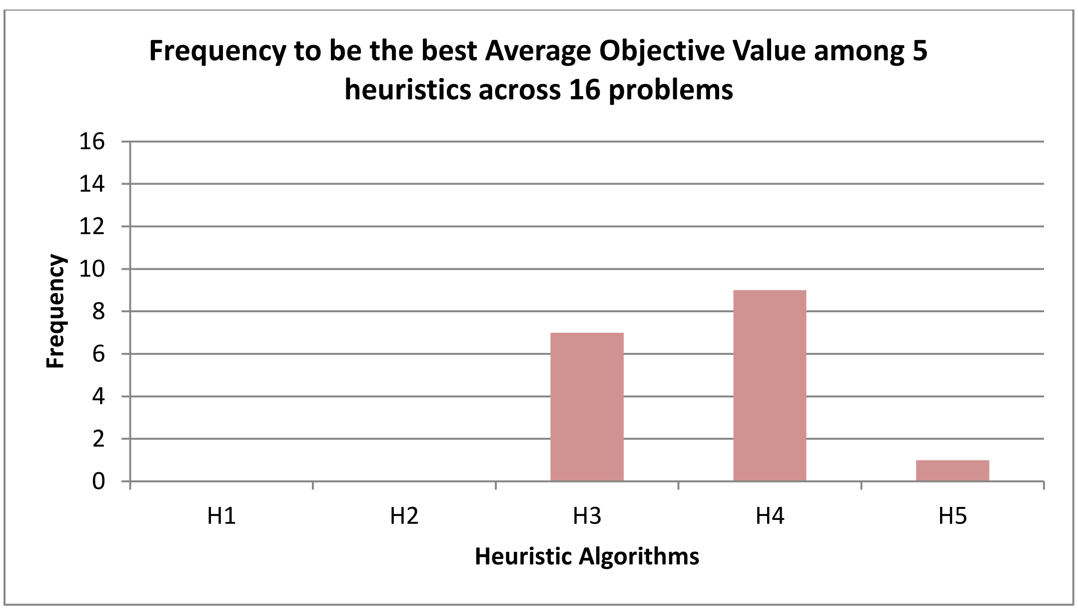

Figure 3.43: Frequency to be the best Average Objective Value among 5 heuristics across 16 problems

As before, the Anderson-Darling test is used to check the normality of the raw data that is reported in Tables B.1 through B.3 in Appendix B. The results as illustrated in the probably plots of Tables B.1, B.2 and B.3 indicate the data does not follow a normal distribution. As such, nonparametric statistical tests are required.

Figure 3.38, 3.39 and 3.40 suggest that a linear relationship exists between the measures of solution quality and the number of customers in system. It is also of interest to determine if measures of solution quality are dependent on the heuristics. As such, a Freidman test is performed to statistically explore these two observations. This test is similar to a two-way ANOVA but is nonparametric.

The 5 heuristics are the first factor in the analysis and number of customers in the system is the second factor. The Freidman test ranks the heuristics for each test problem with the best performance heuristic being assigned rank 1, the second best rank 2, and so on. In case of ties, average ranks are assigned so, for example, if 2 heuristics are tied for rank 2, they will both be ranked 2.5 and next rank is 4. The three Freidman tests are conducted using the hypotheses shown in Table 3.18.

Table 3.18: Hypotheses of Friedman Rank Sum Test for A3 Analysis

A3 Test1	H0: median of the minimum cost is equal for all heuristics H1: at least one is median is different
A3 Test2	H0: median of the maximum objective value of different heuristics are equal H1: not all median of the maximum objective value of different heuristics are equal
A3 Test3	H0: median of the average objective value of different heuristics are equal H1: not all median of the average objective value of different heuristics are equal

Appendix Figure B.4, B.5 and B.6 present the statistical result of the Freidman tests with R. All these tests provides very small *p*-value (<0.001) which means that there is sufficient evidence to reject the null hypothesis and conclude that at least one median value is different for each of the three measures. The post-hoc test results in all three Freidman test results show that the H3 and H4 are significant different with the H1, H2 and H5. As explained above, the lower rank represents the better solution. Table 3.19 and Figure 3.44 gather all heuristics sum of the rank for three Freidman tests. The H3 and H4 have lower sum ranks than the rest three heuristics. Recall the post-hoc analysis results for three Freidman tests indicate the H3 and H4 are significant different with H1, H2 and H3, it can conclude that the H3 and H4 performance better than the other heuristics.

Table 3.19: All Heuristics Summer Rank for Three Freidman Test

	H1	H2	H3	H4	H5
A3 Test 1 (Minimum Objective value)	65	79	31	26.5	38.5
A3 Test 2 (Maximum Objective Value)	54	77	24.5	27.5	57
A3 Test 3 (Average Objective Value)	59	79	26.5	24.5	51

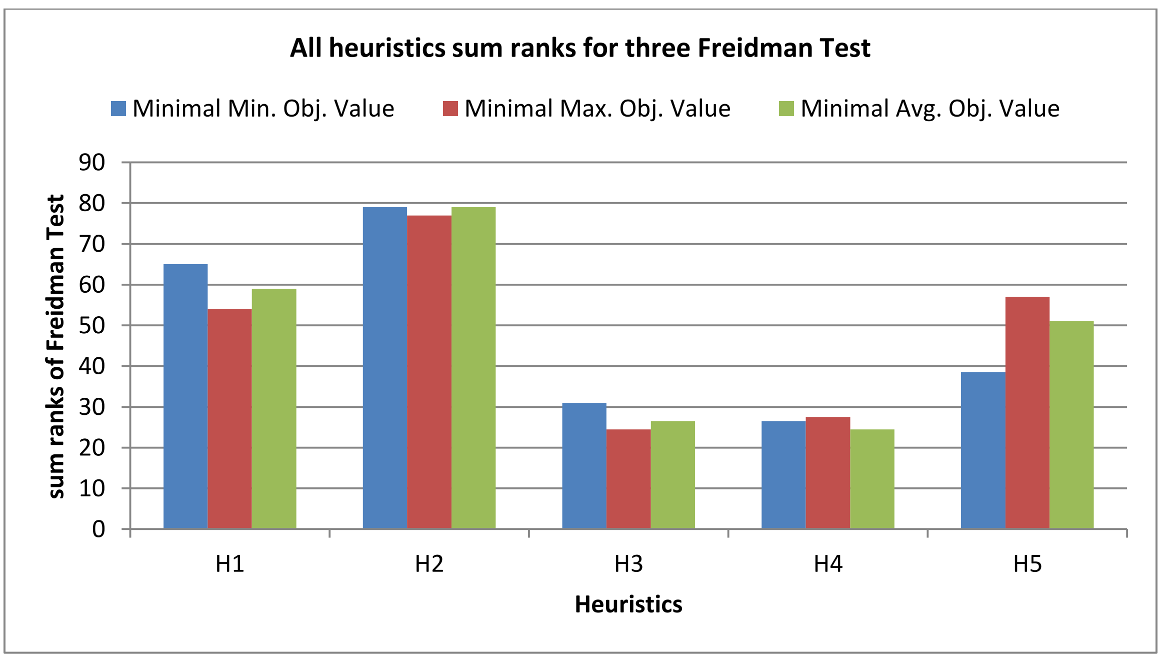

Figure 3.44: All Heuristic Sum Ranks for Three Freidman Test

4. A4 Analysis

In the previous analyses (A1, A2 and A3), the test problems differed only in a few input parameters such as the planning time horizon, number of customers in the system, customer location and demand. The other input parameters were the same. Analysis A4 investigates the sensitivity of the solution to changes in four of these: product lifetime, vehicle fixed cost, vehicle variable cost, and plant product rate. To perform this analysis, a problem with 60 customers is used as the base problem. The input parameters for this test problem are displayed in the Table 3.20. The problems that will be used to perform the sensitivity analysis are generated as follows: 1) Select the input parameter to be varied and fix other input parameters with their value used in the base case. 2) Generate a group of sub-problems by varying the test parameter from its base value. Each sub-problem will vary the input parameter 10 percent from the base value. The range for the parameter under investigation is ±90% of base value. For example, if the product lifetime (B=150) is select as test input parameter, the other input parameter will use the base value in Table 3.20, and a group of 19 sub-problems will be generated. The first will have product lifetime = 150*(1-90%) = 15, the second will have product lifetime = 150*(1-80%)=30, and so on until get the last sub test problem has a lifetime =150*(1+90%)=285. The H4 heuristic is used to solve the problems and each problem will be replicated 50 times. The sensitivity analysis running results for the 4 input parameters with the 60 customer base problem are shown in Tables D.1 through D.3 in Appendix D.

Table 3.20: Base Input Parameters Value for A4 Analysis

Parameters	Values
Customer demand	$q_i \sim U(1, 15)$
Total Number of customer	60
Customer location range	(x, y), $0 \le x \le 100$, $0 \le y \le 100$
Plant location	(50, 50)
Plant production rate	r=1
Vehicle capacity	C_1=12, C_2=15
Vehicle variable cost per unit distance	R_1=1, R_2 =1.1
Vehicle fixed cost	F_1=500, F_2 =600
Product lifetime	B=200
Time Horizon	H=800

Figure 3.45 plots the data from Table C.1 to show the minimum objective function value of system transportation cost by changing 4 input parameters. As shown in the figure, the variable vehicle cost and the fixed vehicle cost have linear relationships with the system transportation cost. Plant production rate and product lifetime have no discernible relationship with transportation cost. To further explore this observation statistically, Analysis A4 uses a correlation analysis to test whether there is correlation between input parameters (variable vehicle cost and the fixed vehicle cost) and objective value. The raw data is checked for normality using Anderson-Darling test and the resulting probability plots are shown in Figures C.4 and C.5. These tests indicate that the data for variable vehicle cost and fixed vehicle cost are following a normal distribution (p-value>0.8). So, 2 Pearson Correlation Tests are conducted to test the correlation between input parameters and objective values. All the hypotheses are shown in Table 3.21.

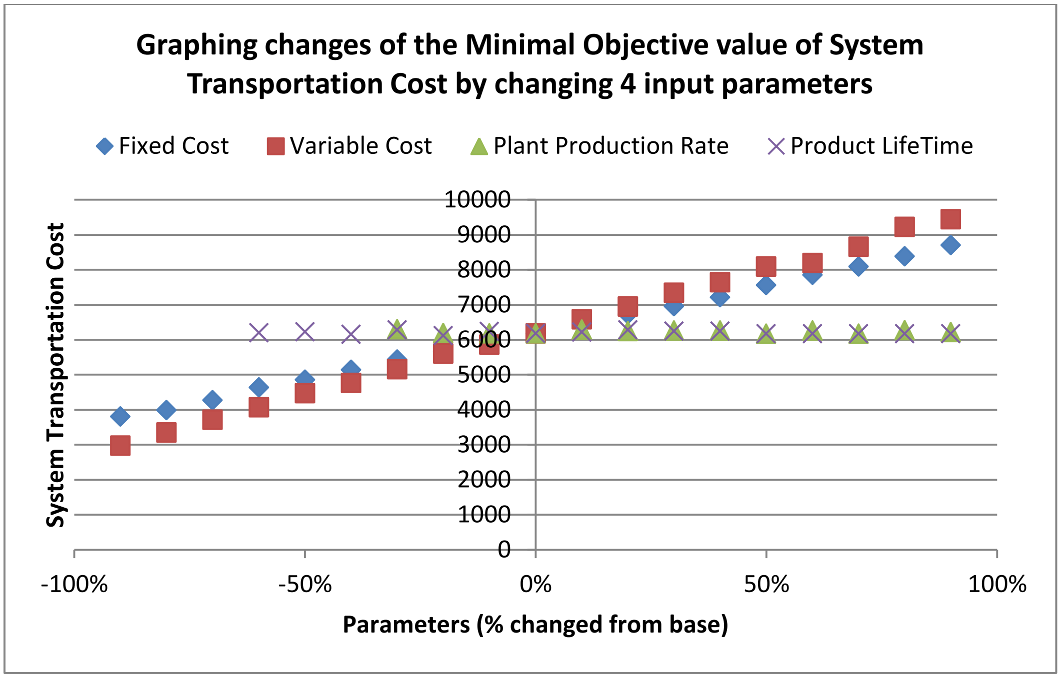

Figure 3.45: Graphing changes of the Minimum Objective value of System Transportation Cost by changing 4 input parameters

Table 3.21: Hypotheses of Tests for A4 Analysis

A4 Test1	H0: the correlation between variable vehicle cost and system transportation cost is 0 H1: the correlation between variable vehicle cost and system transportation cost is not 0
A4 Test2	H0: the correlation between fixed vehicle cost and system transportation cost is 0 H1: the correlation between fixed vehicle cost and system transportation cost is not 0

Both Correlation Test results are shown in Appendix D. The test results indicate that the variable vehicle cost and fixed vehicle cost have a correlation with the system transportation cost (p-value<0.001). Furthermore, the correlation is a strong positive linear relationship (correlation >=0.99).

The linear relationship between the system transportation cost and 2 cost structure input parameters (variable vehicle cost and fixed vehicle cost) is obvious. Changing the fixed vehicle cost and variable cost will change the total system cost. The other 2 input parameters (plant production rate and product lifetime) are more control with the feasibility of the solution. The plant production rate determines whether all customer demand can be satisfied within the time horizon or not and the product lifetime determines the deliverable customer location range size. The Appendix Table C.1 shows that the test problem will not able to get feasible solution when the plant production rate lower than 60% of current base value or the product lifetime lower than 30% of current base value.

Conclusions and Future Research

The IPDSPPP problem had been explored from a number of perspectives in this chapter. A mixed integer model was proposed but could only be solved to optimality for small number of customers because the underlying problem is NP-hard. This fact justifies heuristic approaches for larger problems so several were developed and tested.

Experimental results were obtained using the AMPL and GUROBI 3.0 on a Dell Optiplex 755 with 2.33GHz Intel Core 4 Duo CPU and 4096 gigabytes dual channel memory. The mathematical programming model could be solved to optimality for 7 customers but no more. As a transition to heuristic approaches, a new approach to get LB was proposed for the problem. The new approach of the LB had been shown to be statistically superior to the existing one.

The heuristic approach taken to IPDSPPP in this work uses a simulated annealing search algorithm based structure. Several different versions within this basic structure were developed and applied to different scenarios of the IPDSPPP to find approximate solutions. A number of experimental comparisons were performed with interesting results. One was that the Multi-stop Delivery with Random Shortest Path Fit algorithm with the simulated annealing search method (H4) and Multi-stop Delivery with Shortest Path Fit algorithm with the simulated annealing search method (H3) are statistically superior to the other heuristic algorithms in all 3 test criteria (i.e. minimum cost, maximum cost and average value of the system transportation cost). Furthermore, the series of experiments for the sensitivity analyses of the rest input parameters (i.e. vehicle fixed cost, vehicle variable cost, plant production rate, product lifetime) show that the vehicle fixed cost and vehicle variable cost have the positive linear effect on the total system transportation cost. The product lifetime and plant production rate are more likely to control the feasibility of the solution rather than to affect the total system transportation cost.

There are several interesting ways that the IPDSPPP problem can be expanded:

1. There is obviously new ways that the customers capable of being served by multiple plants can be partitioned. In addition to this, a practical aspect to explore would be comparing the additional computational burden of more complex procedures to the quality of the solution.

2. Split delivery should be investigated. A first step would be to allow the demand at a single customer to be satisfied by 2 or more vehicles. Then, 2 or more vehicles from different plants can be the topic of an interesting research project.
3. Adding time windows when customers can accept shipments is another important feature that could be added which would increase the realism dramatically for many practical situations.

CHAPTER FOUR

MULTIPLE PLANTS INTEGRATED PRODUCTION AND DISTRIBUTION PROBLEM WITH SINGLE PERISHABLE PRODUCT

Introduction

In Chapter 3, a mathematical model is proposed for the single plant integrated production and distribution problem with a perishable product problem (IPDSPPP) with the objective of reducing system transportation cost while maintaining an acceptable level system performance. The model considered a network with one centralized plant and multiple geographically separated customers and the decisions were the production and distribution schedules to satisfy all customer demands within the product's lifetime. All customers are located so that delivery within this time is possible with vehicles that are available (i.e., a feasible solution exists) and the research is based on single period. Optimal solutions were obtained for the problem with small datasets that included less or equal to 7 customers in the system and several heuristics were developed to find approximate solutions.

The research reported in this chapter extends the basic idea from the previous chapter to multiple plants. This is certainly a reasonable extension from a practical viewpoint because two geographically separated plants not only provide additional capacity to serve customers in the region that they can both reach within the time horizon, but it also extends the sales region as well. Clearly, if several businesses have large-sized orders, the total customer demand could easily overwhelm the capacity of the single plant capacity. The perishable property of the product along with the time horizon are not only important from a modeling perspective, they are genuinely restrictive in practice. Defining the plant's deliverable range as the area that customer demand can be served by the plant with a product before the lifetime has expired, it is easy to think of the deliverable range as the area of a circle that has the plant located at the center and a vehicle's maximum driving distance within the product lifetime as the radius. This correctly reflects the fact that the delivery range is proportional to the product lifetime and, as such, there are at least three ways to increase the deliverable range. The first would be to extend the product lifetime. Some types of perishable products will have a longer lifetime with different storage methods. For example, it is easy to visualize that the delivery range for some types of chemical or biological substances could be dramatically enlarged with refrigerated or pressurized transport. Relative to this research, this approach violates a basic tenet, namely, that vehicles are readily available for hire in any quantity because these transporters are very specialized. While this is a terrific option in practice for certain perishable products, it is not of interest in this research. The second method would be to increase the speed of the vehicle. Again, this is completely reasonable for certain types of perishable products but of no interest in this research. The point is that there are clearly other techniques that could be used in practice to extend the range of a company delivery area; however, this research focuses on the situation where the choice is to open a second production facility. In addition to extending the range, this method can also decrease the average distance from the customer to the closest plant center which might lead to a reduction of the total delivery cost as well.

Literature Review

The current problem with multiple plants could be considered an extension of the Multiple Depot Vehicle Routing Problem (MDVRP). After Clarke and Wright (1964) proposed the basic VRP problem in 1964, many extensions have been proposed. The MDVRP problem is motivated by economics and practicality. Laporte et al. (1988) used an appropriate graph representation and transformed the MDVRP problem into an equivalent constrained assignment problem. They provided a branch-and-bound method to solve the optimal solution up to 80 nodes in the system. Sumichrast and Markham (1995) developed a heuristic method to solve the MDVRP problem for distributing raw materials between multiple resources and multiple plants. They adapted the Clarke and Wright saving method (1964) to create their algorithm and measured its performance by comparing it with a lower bound. Several other heuristics have been proposed to solve the MDVRP. Renaud et al. (1996) adopted a tabu search that initially assigns all customers to their nearest plant then and improves the solution with tabu search. Salhi and Sari (1997) proposed a heuristic method with three levels that first constructed an initial feasible solution and, then, improved the routes from each depot in the next two steps alternative repeating applying the local search technique and different composite heuristic until hit the stop criteria. In more recent years, Ho et al. (2008) proposed a hybrid genetic algorithm for the MDVRP. The algorithm generates initial solutions by two methods: 1) randomly and 2) an adaptation of Clarke and Wright's saving method (1964) and a nearest neighbor heuristic. All of these models assume that each vehicle makes only one trip in the planning horizon. The VRP in which vehicles can make multiple routes during the planning horizon (or called multiple uses of vehicles) is another important extension of basic VRP. The first work to address multiple uses of vehicles into VRP problem was Salhi (1987); this feature added significant realism and practicality to the basic problem. Later, Taillard et al. (1996) provide a tabu search algorithm to solve the vehicle routing problem with multiple uses of vehicles. It uses a bin packing algorithm to assign the routes to vehicles. Brandao and Mercer (1997) proposed a three-phase heuristic which starts with an insertion heuristic for initial solution, then the solution is improved using a tabu search by reinsertion and exchange of customers, feasibility of the solution is not considered in this phase. The third phase is to restore the search limit while only considering the feasible solution. In a second paper by Brandao and Mercer (1998), a more complex variant of the problem with maximum overtime constraints and a mixed fleet is considered. More recently, Petch and Salhi (2003) proposed a three-phase heuristic to solve a variant of the problem where, for a given number of vehicles, the objective is to minimize the maximum overtime. Their population-based approach first generates routes with a savings-based heuristic. These routes are then combined to form complete solutions which are finally improved with a local search heuristic. Olivera and Viera (2007) presented an adaptive memory-based heuristic, where the memory is made of multiple route solutions.

The literature that considers multiple depots and multiple use vehicles is scarce. It is even less when considering the perishable product in the system. The only literature we could find in the similar area is the work of Devapriya (2008). He proposed the research with a single plant, multiple identical vehicles, multiple customers and a fixed time horizon. The research assumed that the vehicles used for distribution can make multiple trips as long as all orders are satisfied within the planning horizon. The objective is to determine the minimum fleet size, their route and the production schedule to minimize the total system distribution cost. Two heuristics are proposed to solve this strongly NP-hard problem.

One difference between this research and that of Devapriya (2008) is the use of multiple types of non-identical vehicles for distribution. Each type of vehicle has the unique capacity, variable running cost and fixed running cost. Another is that this research considers multiple plants - it is the multiple plant integrated production and distribution schedule problem with single perishable product (MPIPDSPPP).

In the following sections of this chapter, a mixed integer programming model of this problem is proposed as well as heuristic approaches to find approximate solutions to this NP-hard problem. Numerical examples are then presented to illustrate various features of the model and the heuristics.

Problem Description

The MPIPDSPPP problem is an extension of the previous research of the IPDSPPP problem. Here, more than one plant is available to supply the perishable product to customers that are geographically dispersed in an area. Each plant has a fixed production rate and is geographically separated from the other plants. Every plant uses its own fleet of vehicles to deliver the customer demands which means that all vehicles return to the same plant to which they are initially assigned for replenishment. The goal is to satisfy all customer demands within the time horizon and product lifetime while minimizing the total system transportation cost. To achieve this goal, the delivery schedule and production schedule must be determined simultaneously which includes the partition of customers for each plant, number and type of vehicles for each plant, the delivery routes for each vehicle and their dispatch schedule, and each plant's production schedule for each batch. The detailed assumptions for the problem are as follows:

- The performance of the whole supply chain is considered within one planning horizon.
- M different production centers (plants) serve all customers which are geographically separated and independent of each other.
- All customer demands are deterministic and must be satisfied within the time horizon.
- Every plant has an individual fixed production rate.
- A single perishable product is produced and distributed to customers.
- The product has a constant lifetime that begins immediately after the production.
- Expired product cannot be used to satisfy the customer demand. Also, there is no salvage value for products after the lifetime has expired.
- Every plant hires its own delivery fleet and vehicles are assigned to only one plant. That is, a vehicle assigned to Plant 1 can never have any transactions with a plant other than Plant 1.

- The fleet is selected from a portfolio of non-identical vehicles which have different capacities, different unit distance travel costs, and different fixed setup costs.
- There are an unlimited number of each type of vehicle from which the fleet can be selected.
- For each vehicle, deliveries begin and end at the same plant.
- Multiple trips for vehicles are allowed but it must be from the same plant.
- If the vehicle finishes a delivery trip early, it can be assigned to another delivery trip as long as the second trip is feasible.
- Loading and unloading times are not considered.
- Split delivery is not allowed. That is, demand at each customer must be satisfied by one vehicle during one stop; it cannot be satisfied by two or more vehicles or during multiple stops of one vehicle.

In this research, multiple plants are considered. Because of the perishable property of the product, this work is fundamentally different with the traditional MDVRP. Figure 4.1 illustrates an simple example of the MPIPDSPPP problem with two plants. All customer demands can be satisfied by one or both of these two plants. Notice that neither plant in this figure can reach every customer which is a differentiating feature between this research and the classical MDVRP problem. This feature has an important conceptual implication as well. It means that all customers fit into one of three types: Type I - customers who can only be served by the plant 1, Type II - customers who can only be served by the plant 2, and Type III - customers who can be served by plant 1 or plant 2.

Note the special situation for the MPIPDSPPP where there are no Type III customers. Although, this is technically a multi-plant situation, in reality it is simply two IPDSPPP. Since there are no customers located in the overlapping deliverable area, all customers in this scenario can only be served by Plant 1 or Plant 2, so, the current MPIPDSPPP problem can be simplified into 2 independent IPDSPPP problems and solved using the previous research results on IPDSPPP problem. The MPIPDSPPP problem to be addressed in this chapter is the scenario with customers located in the overlapped deliverable range. In the other words, problems must include Type III customers.

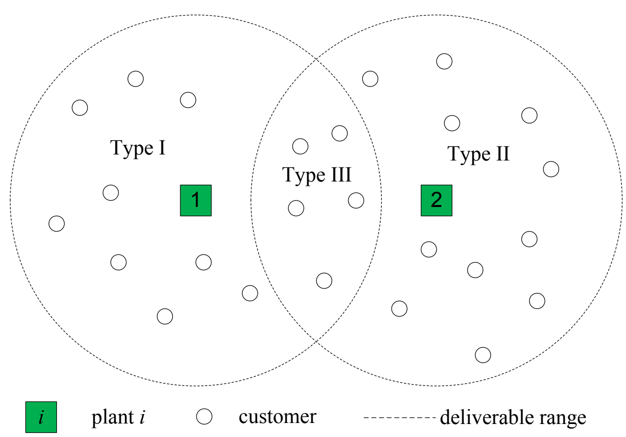

Figure 4.1: An Example of MPIPDSPPP Problem

Mixed Integer Mathematical Model for the MPIPDSPPP

In this section, a mixed integer programming model is proposed for the MPIPDSPPP. Because all customer demands are deterministic and must be satisfied within the time horizon, the total number of products that need to be produced is predetermined and the raw material cost is fixed. Hence, the goal of the model is to minimize the total system transportation cost and the objective function has two major parts. The first part is the fixed cost that is analogous equivalent to a fixed rental fee plus fixed labor cost to operate the vehicle. It is assessed on each vehicle used to make deliveries and does not depend on distance traveled. The second part is the variable cost which is assessed based on the distance traveled by the vehicle during the delivery process.

Mathematical Model

Since the model needs to calculate both the travel time and travel distance for each trip, it is assumed that all vehicles maintain a constant traveling speed at 1 unit distance / unit time. With this assumption, the distance and travel time are the same in value between any two points yet this assumption does not restrict the generality of the model. Before presenting the model, the notation to be used is identified.

Indices:

V = Set of all plants. $V = \{1, 2, ..., m\}$

N' = Set of all customers. $N' = \{m+1, ..., m+n+1\}$

N = Set of all customers and the plants. $N = \{1, 2, ..., m+n+1\}$

S = Set of all type of vehicles. $S = \{1, 2, ..., l\}$

Parameters:

B = Fixed lifetime of the product

H = Time horizon

n = Total number of customers.

m = Total number of plants.

l = Total number of types of vehicles.

q_i = demand of customer i, $i \in N'$

r_v = production rate of the v^{th} plant. $v \in V$

τ_{ij} = Travel time from customer i to customer j, $i, j \in N$

C_s = Capacity of the s^{th} type truck, $s \in S$

F_s = Fixed cost associated with the s^{th} type truck, $s \in S$

R_s = ratio of traveling cost per unit time for s^{th} type of truck to the standard truck. where $s \in S$.

Decision Variables:

$X^{v}_{s\,i\,j\,k\,m}$ = 1 if v^{th} plant's m^{th} truck in s^{th} type visits customer j immediately after customer i in its k^{th} trip, 0 otherwise. $v \in V, s \in S, i, j \in N, k, m \in N'$.

$L^{v}_{s\,k\,m\,t}$ = 1 if the v^{th} plant is producing for its k^{th} trip of m^{th} truck in s^{th} type at time epoch t, 0 otherwise. $v \in V, s \in S, k, m \in N', t \in T$.

$y^{v}_{s\,m}$ = 1 if the v^{th} plant's m^{th} truck in s^{th} type is used for delivery, 0 otherwise. $v \in V, m \in N', s \in S$.

$Z^{v}_{s\,k\,m}$ = 1 if the k^{th} trip of the v^{th} plant's m^{th} truck in s^{th} type is used, 0 otherwise. $v \in V, s \in S, k, m \in N'$.

$d^{v}_{s\,k\,m}$ = Distribution start time of the k^{th} trip of the v^{th} plant's m^{th} truck in s^{th} type. $v \in V, s \in S, k, m \in N'$.

$p^{v}_{s\,k\,m}$ = Production start time of the k^{th} trip of the v^{th} plant's m^{th} truck in s^{th} type. $v \in V, s \in S, k, m \in N'$.

$$Min \quad \sum_{s\in S}\sum_{v\in V}\sum_{i\in N}\sum_{j\in N}\sum_{k\in N'}\sum_{m\in N'}\left(X^{v}_{s\,i\,j\,k\,m}\cdot R_s \cdot \tau_{i\,j}\right)+\sum_{v\in V}\sum_{s\in S}\sum_{m\in N'}\left(F_s \cdot y^{v}_{sm}\right)$$

$Subject\ to:$

$$\sum_{i\in N}\sum_{j\in N} X^{v}_{s\,i\,j\,k\,m}\cdot q_j \le C_s \qquad \forall\, v\in V, s\in S, m,k\in N' \tag{1}$$

$$\sum_{s\in S}\sum_{v\in V}\sum_{i\in N}\sum_{k\in N'}\sum_{m\in N'} X^{v}_{s\,i\,j\,k\,m} = 1 \qquad \forall\, j\in N' \tag{2}$$

$$\sum_{s\in S}\sum_{v\in V}\sum_{j\in N}\sum_{k\in N'}\sum_{m\in N'} X^{v}_{s\,i\,j\,k\,m} = 1 \qquad \forall\, i\in N' \tag{3}$$

$$\sum_{i\in N} X^{v}_{s\,i\,j\,k\,m} = \sum_{i\in N} X^{v}_{s\,j\,i\,k\,m} \qquad \forall\, v\in V, s\in S, m,k,j\in N' \tag{4}$$

$$\sum_{j\in N'} X^{v}_{s\,v\,j\,k\,m} = Z^{v}_{s\,k\,m} \qquad \forall\, v\in V, s\in S, m,k\in N' \tag{5}$$

$$\sum_{i\in N'} X^{v}_{s\,i\,v\,k\,m} = Z^{v}_{s\,k\,m} \qquad \forall\, v \in V, s \in S, m, k \in N' \tag{6}$$

$$e_i - e_j + 1 \le n \cdot \left(1 - X^{v}_{s\,i\,j\,k\,m}\right) \quad \forall\, v \in V, s \in S, j, k, m \in N', \tag{7}$$

$$\sum_{s\in S} y^{v}_{s\,m} \le 1 \qquad \forall\, v \in V, m \in N' \tag{8}$$

$$X^{v}_{s\,i\,j\,k\,m} \le y^{v}_{s\,m} \qquad \forall\, v \in V, s \in S, i, j \in N, k, m \in N' \tag{9}$$

$$X^{v}_{s\,i\,j\,k\,m} \le Z^{v}_{s\,k\,m} \qquad \forall\, v \in V, s \in S, i, j \in N, k, m \in N' \tag{10}$$

$$d^{v}_{s\,k\,m} - \left(p^{v}_{s\,k\,m} + \frac{1}{r_v} \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot q_j \right) + \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot \tau_{i\,j} \le B \quad \forall\, v \in V, s \in S, k, m \in N' \tag{11}$$

$$d^{v}_{s\,k\,m} + \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot \tau_{i\,j} \le H \qquad \forall\, v \in V, s \in S, k, m \in N' \tag{12}$$

$$p^{v}_{s\,k\,m} + \frac{1}{r} \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot q_j \le d^{v}_{s\,k\,m} \qquad \forall\, k, m \in N', \quad v \in V, s \in S \tag{13}$$

$$d^{v}_{s\,k\,m} + \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot \tau_{i\,j} \le d^{v}_{s\,k+1\,m} \qquad \forall\, v \in V, s \in S, k, m \in N' \tag{14}$$

$$p^{v}_{s\,k\,m} + \frac{1}{r} \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot q_j \le p^{v}_{s\,k+1\,m} \qquad \forall\, v \in V, s \in S, k, m \in N' \tag{15}$$

$$\sum_{s\in S} \sum_{k\in N'} \sum_{m\in N'} L^{v}_{s\,k\,m\,t} \le 1 \qquad \forall\, v \in V, t \in T \tag{16}$$

$$\sum_{t\in T} L^{v}_{s\,k\,m\,t} \ge \frac{1}{r_v} \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot q_j \qquad \forall\, v \in V, s \in S, k, m \in N' \tag{17}$$

$$p^{v}_{s\,k\,m} \le t \cdot L^{v}_{s\,k\,m\,t} + M \cdot \left(1 - L^{v}_{s\,k\,m\,t}\right) \qquad \forall\, v \in V, s \in S, t \in T, k, m \in N' \tag{18}$$

$$p^{v}_{s\,k\,m} + \frac{1}{r} \sum_{i\in N} \sum_{j\in N'} X^{v}_{s\,i\,j\,k\,m} \cdot q_j \ge t \cdot L^{v}_{s\,k\,m\,t} \qquad \forall\, v \in V, s \in S, t \in T, k, m \in N' \tag{19}$$

$$X^{v}_{s\,iikm} = 0 \qquad \forall\, k, m \in N', \quad i \in N, \quad v \in V, s \in S \tag{20}$$

$$X^{v}_{s\,i\,j\,km} = 0 \quad where\; i \neq v, \quad \forall\, k, m, j \in N', \quad i, v \in V, s \in S \tag{21}$$

$$X^{v}_{s\,i\,j\,km} = 0 \quad where\; j \neq v, \quad \forall\, k, m, i \in N', \quad j, v \in V, s \in S \tag{22}$$

Constraint (1) limits the total customer demand in a given trip so that it cannot exceed the vehicle capacity. Constraints in (2), (3) and (4) ensure that each customer's demand will be met and met only once. Constraints (5) and (6) ensure that every delivery trip begins and ends at the plant. Constraint (7) is a rather standard sub-tour elimination approach. Constraints (8), (9) and (10) restrict each vehicle to be of only one type and enforce this in the route-defining variables. Constraint (11) ensures that deliverer are completed before the product lifetime expires and (12) forces every delivery to be completed before the time horizon ends. Constraints (13) and (14) ensure that a delivery cannot begin before the product is ready and that the delivery truck cannot start a new trip until it has returned to the plant from its previous trip. Constraint (15) forces production in the sequence of delivery and forbids preemption. Constraint (16) restricts the plant to produce for one trip in any time period and (17) ensures each plant produces with its fixed productivity. Constraint (18) and (19) together ensure the plant continues to produce one batch until the entire quantity required for a trip is completed. Constraint (20) prevents a vehicle delivery the product to itself. Constraints (21) and (22) ensure the vehicle always return back to the plant it start with.

Table 4.1 and Figure 4.2 show the growth of the problem size as a function of an increasing number of the customers. 10 problems with 3 to 16 customers are conducted for the comparison. Recall that the MIP Gap refers to the gap between the mixed integer lower bound given by the GUROBI and the current best integer solution,

$$MIP\ gap = \frac{Current\ Best\ Integer\ Solution - MIP\ Lower\ Bound\ Given\ by\ Gurobi}{Current\ Best\ Integer\ Solution}$$

The default stopping criterion is to terminate the program when the MIP gap is less than 0.0001. The testing is conducted using AMPL and Gurobi3.0 with a Dell workstation computer (Dell Optiplex 755; 2.33G Hz Intel Core 4 Duo CPU with 4096 gigabytes dual channels memory).

Table 4.1: Growth of the Problem Size with the Number of Customers in MPIPDSPPP

Problem Number	Number of Customers	Number of Variables	Number of Constraints	MIP Gap (%)	Gurobi Solution	Approximate Elapsed Time (sec)	Stop Criteria
1	3	7866	16416	0	736	10	Default
2	4	14488	30362	0	800	5	Default
3	5	23630	50132	0	1322	185	Default
4	6	35748	77070	0	949.8	84902	Default
5	7	51394	112808	0	1516	42929	Default
6	8	67376	151330	36.7	1516.8	67881	Out of memory
7	9	112158	251152	30.3	1479	46650	Out of memory
8	10	142460	325362	n/a	n/a	39900	Out of memory
9	14	383460	920822	n/a	n/a	n/a	Out of memory
10	16	560224	1382778	n/a	n/a	n/a	Out of memory

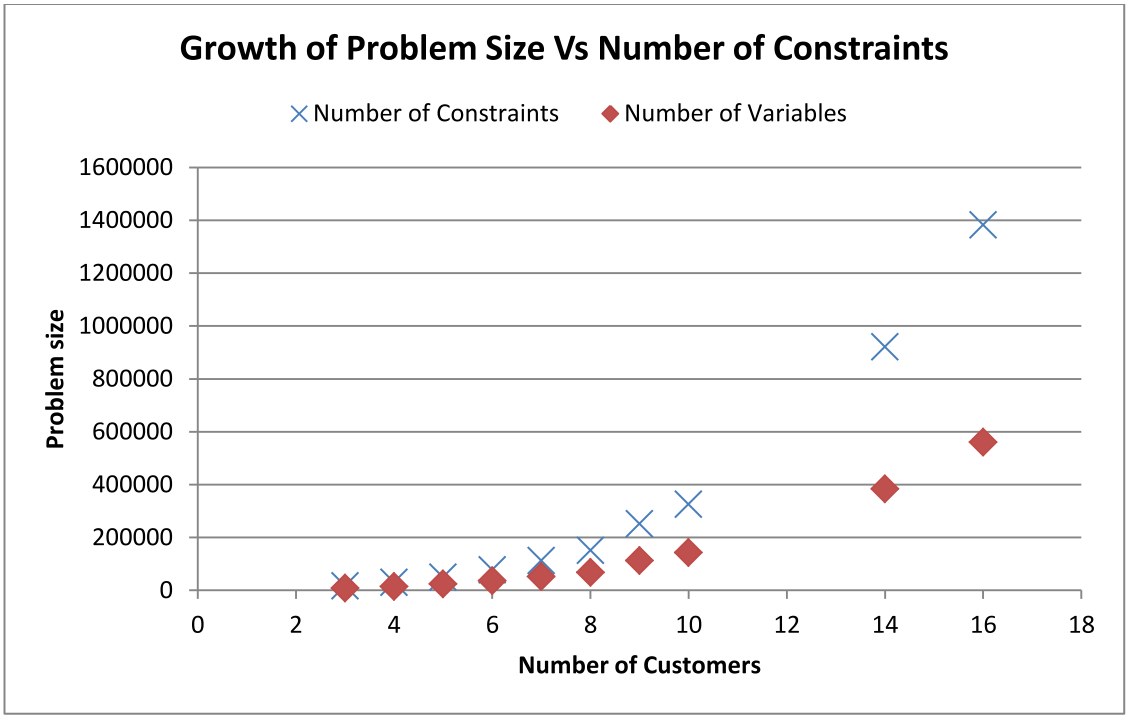

Figure 4.2: Growth of Problem Size of MPIPDSPPP with Number of Customers

The MPIPDSPPP problem is an extension of MDVRP problem which had been proven to be an NP-hard problem, so the MPIPDSPPP inherits the complexity of MDVRP and is also NP-hard. As shown in the figures, even though AMPL and GUROBI can solve the mixed integer programming model for MPIPDSPPP optimally, the number of constraints and decision variables grow quickly when the number of customer in the system increased. The optimal solution only can be obtained when the problem has 7 or fewer customers in the system.

Heuristic Algorithms for the MPIPDSPPP

The MPIPDSPPP problem is NP-hard and the practical reality of this is reflected in the numerical examples of the previous section. Table 4.1 shows that finding the optimal solution for the MPIPDSPPP problem with 2 plants and 7 customers takes over several hours (elapsed time). Further, a feasible solution cannot be obtained when the problem size increases to 10 customers. The fact that MPIPDSPPP is NP-hard justifies heuristic approaches to find approximate solutions for larger problems. Several heuristics are now proposed for the MPIPDSPPP problem.

Structure of the Heuristic Algorithms

Recall previous introduction, the MPIPDSPPP problem is an extension of the IPDSPPP problem that increases the number of plants in the planned area. As shown in the Figure 4.1, that MPIPDSPPP problem can be viewed as two IPDSPPP problems if the customers able to be served by either plant are assigned to one of the plants. On an intuitive level, assigning customers then becomes the crux of any heuristic and it can have a dramatic impact on the solution. With this thought, the basic heuristic algorithm decision hierarchy is presented in Figure 4.3. It can be seen that, first, the customers in the overlapping region are assigned to different plants (i.e. the customer allocation result is generated), then the IPDSPPP problem is solved for each plant (Step 3 to 5) separately, and the last solution result is found by adding the solutions together. This idea drives that basic structure of this heuristic:

Step 1: Assign all customers to plants.

Step 2: In each plant, assigning initial plant production schedule.

Step 3: In each plant, assigning customers to routes.

Step 4: In each plant, assigning routes to vehicles and refine the production schedule.

Step 5: Merge all plant production schedules and vehicle routes into a last solution.

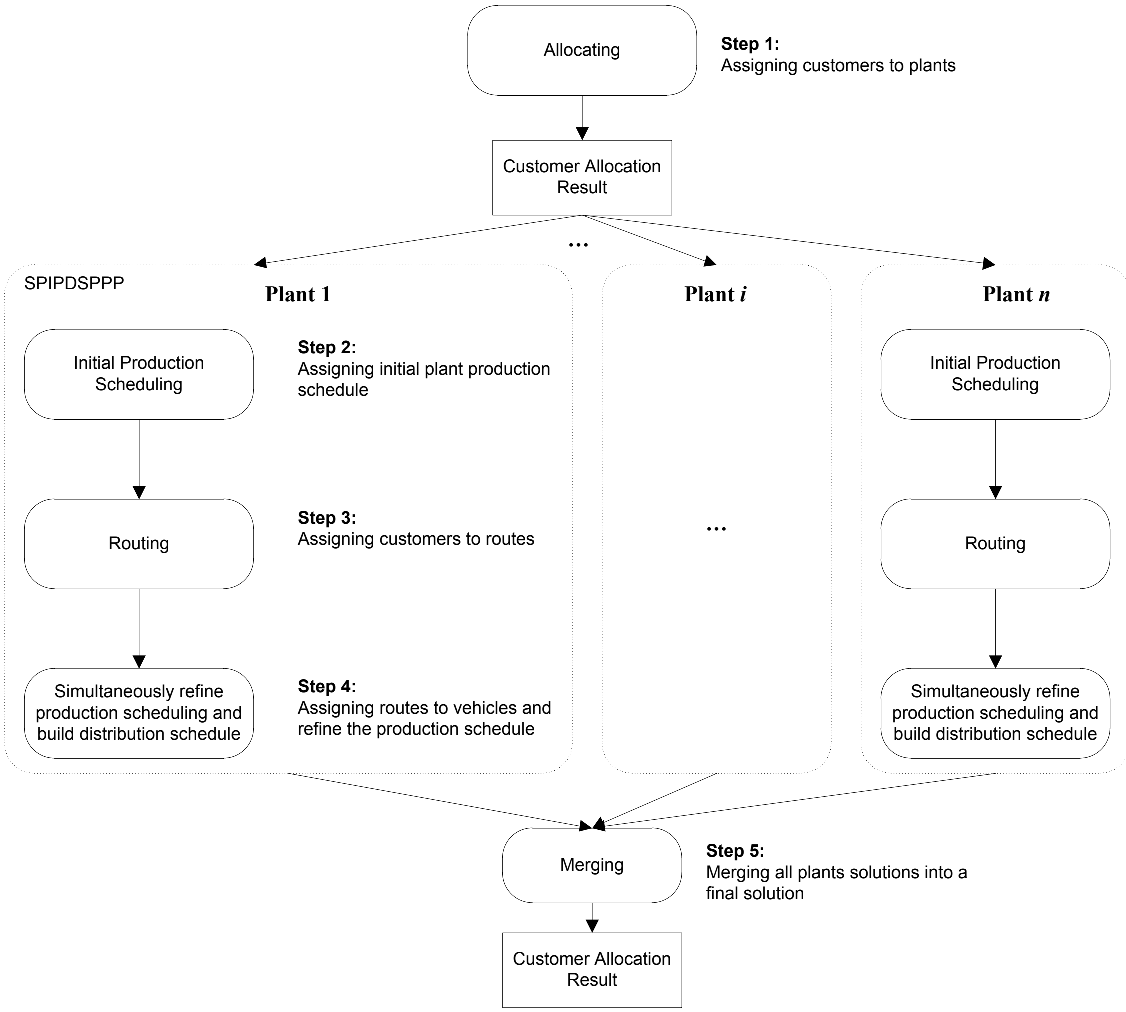

Figure 4.3: The Basic Hierarchy of Decisions in the MPIPDSPPP

The challenge is Step 1 because Steps 2, 3 and 4 comprise the work done previously on IPDSPPP and merging non-overlapping plants solutions (Step 5) is straight forward. The key is to determine how to assign customers of Type III to one of the two plants because this places a bound on the quality of the solution. If customers are poorly assigned, for example, a much larger number of direct deliveries are required than the other customers assignment; then the transportation cost of the solution can never bit the solution which had better customer assignment. Also important is the fact that inappropriate assignments are not correctable in the later steps. Simply stated, since it is assumed that each plant hires their own fleet for the delivery, if the initial partition assigns Customer A to Plant 1 and the optimal solution has Customer A assigned to Plant 2, the solution can never reach optimality, or possibly be relegated to be rather poor and this is determined before any algorithm is used. So, the quality of the solution result is highly dependent on the initial assignments and resulting customer partitions.

Complexity of Customer Allocation

It has been argued that the effectiveness of customer allocation step will affect the quality of the MSIPDSPPP problem. This is different from the customer allocation in classical MDVRP problem because it might not be possible to assign all customers to any plant because of product lifetime or horizon constraints. The general impact this has on the heuristics is that the feasibility of the customer allocation should be guaranteed. The infeasible customer allocation always leads to an infeasible last solution result and cannot be fixed in later steps. To facilitate the discussion of the customer allocation problem, it is categorized into two cases: 1) Customer allocation when there is one overlapping region so all plants can serve every customer in the overlapping region (CL1) and 2)

Customer allocation when there are multiple overlapping regions so all plants cannot serve all customers in the overlapping regions (CL2).

Customer Allocation When There Is One Overlapping Region So All Plants Can Serve Every Customer in the Overlapping Region (CL1)

Example 4.1 is used to illustrate partitioning of customers when there is one overlapping region so that all customers in the overlapping region can be served by all plants. This is the only overlapping region for two plants as shown in Figure 4.4 but it can also be true for multiple plants as well.

Define the following:

A: a set of all customers

Θ: a set of plants

A_Θ: disjoint service set; this is the set of customers that can be and only can be served by plants in the set Θ. In Figure 4.4, $A_{\{1,2\}} = \{1,2,3\}$ means the customer 1, 2 and 3 only can be served by either plant 1 or plant 2.

$Out(A_\Theta)$: It is the collection of all possible allocations of customers in set A_Θ to plants in set Θ.

$CN(A_\Theta)$: total number of possible allocations in set $Out(A_\Theta)$, it is $|Out(A_\Theta)|$

$CN(A)$: total number of possible allocations for customer in set A.

Example 4.1: Consider 6 customers and 2 plants in the planned area as shown in the Figure 4.4. The means that $A_{\{1\}}=\{4\}$, $A_{\{2\}}=\{5\}$, and $A_{\{1,2\}}=\{1,2,3\}$.

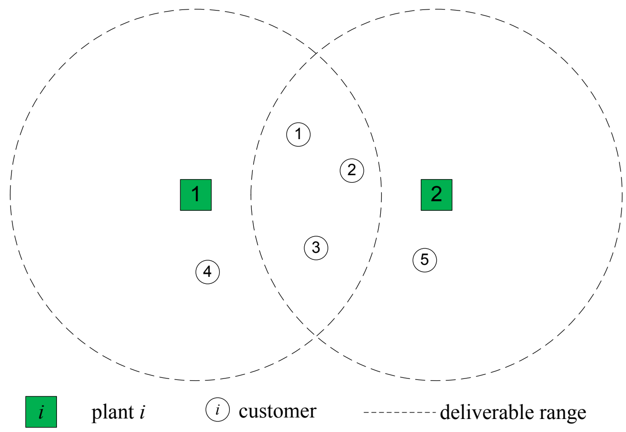

Figure 4.4: Customers and Plants Location Graph for the Example 4.1

As shown in Figure 4.4, customers 4 and 5 are in the non-overlapping region only can be served by one plant. Their allocation is fixed and will not change with any allocation method. So customer allocation methods focus exclusively on the customers in the overlapping region or regions, in this example it is set $A_{\{1,2\}}$. Define R_i as the set of customers in A_Θ who are assigned to plant $i\in\Theta$. In this example, we have R_1 and R_2 that will denote the customers in $A_{\{1,2\}}$ that are assigned to plants 1 and 2, respectively. Note that $|A_{\{1,2\}}| = |R_1| + |R_2|$. Table 4.2 shows all possible allocation results for $Out(A_{\{1,2\}})$ using the data in Example 4.1. The total number of possible allocations can be determined by combinatorics. Choose the customers for set R_1 first and then, assign the rest of the customers to set R_2. The number of customers that could be assigned to set R_1 ranges from 0 to $|A_{\{1,2\}}|$=3 so the total number of allocations for the example is:

$$CN(A_{\{1,2\}}) = \binom{3}{0}\cdot\binom{3}{3} + \binom{3}{1}\cdot\binom{2}{2} + \binom{3}{2}\cdot\binom{1}{1} + \binom{3}{3}\cdot\binom{0}{0} = 8.$$

Table 4.2: All possible Allocation Result for A_Θ for Example 4.1

No.	R_1	R_2	$[\lvert R_1\rvert, \lvert R_2\rvert]$
1	∅	{1,2,3}	[0,3]
2	{1}	{2,3}	[1,2]
3	{2}	{1,3}	
4	{3}	{1,2}	
5	{1,2}	{3}	[2,1]
6	{1,3}	{2}	
7	{2,3}	{1}	
8	{1,2,3}	∅	[3,1]

Proposition 4.1: The number of customer allocations for the scenario with 2 plants and n overlapping customers is 2^n.

Proof: Generalizing the method for Example 4.1, we get the following expression for $CN(A_{\{1,2\}})$ with $|A_{\{1,2\}}| = n$.

$$CN(A_{\{1,2\}}) = \binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{n} \tag{23}$$

This can be simplified using the binomial theorem formula

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k \tag{24}$$

Setting $x = 1$ and $y = 1$ yields $CN(A_{\{1,2\}}) = 2^n$.

The result of Proposition 4.1 shows that the total number of allocations $CN(A_\Theta)$ is dependent on the number of customers in the overlapping region, $|A_\Theta|$ = n, and the number of plants available to serve these customers, $|\Theta| = m$. So, define

$$CN(n, m) = CN(A_\Theta), \text{ where } |A_\Theta| = \text{n, and } |\Theta| = m \tag{25}$$

to be the total number of allocations for the set of overlapping customers A_{Θ},. Clearly, $CN(0, i) = 1$, $CN(i, 1)=1$, where $i \geq 1$.

Proposition 4.2: The number of allocations for the scenario with m plants and n overlapping customers is m^n.

Proof: $|A_{\{1,2,\ldots,m\}}| = n$ customers in set $A_{\{1,2,\ldots,m\}}$ that are in one overlapping region must be assigned to m plants. Each plant is assigned customers in sequence, $R_1 \rightarrow R_2, \ldots, \rightarrow R_m$. Plant 1 is assigned i customers that are placed into set R_1. The remaining problem is to allocate n-i customers to m-1 plants which is $CN(n\text{-}i, m\text{-}1)$. The general number of allocation is:

$$CN(n,m) = \binom{n}{0} \cdot CN(n,m-1) + \binom{n}{1} \cdot CN(n-1,m-1) + \ldots + \binom{n}{n} \cdot CN(0,m-1) \qquad (26)$$

Proof is by Mathematical Induction.

1. $CN(n, m) = m^n$ holds when m=1, n=1; that is, $CN(1, 1) = 1$. This is obvious since 1 customer and 1 serviceable plant only has one allocation.

2. Assume as the induction hypothesis $CN(n, m) = m^n$.

3. Prove that the hypothesis is true for $CN(n+1, m)$ and $CN(n, m+1)$. From (26) and (24), the following equations are generated:

$$\begin{aligned} CN(n,m+1) &= \binom{n}{0} \cdot CN(n,m) + \binom{n}{1} \cdot CN(n-1,m) + \ldots + \binom{n}{n} \cdot CN(0,m) \\ &= \binom{n}{0} \cdot m^n \cdot 1^0 + \binom{n}{1} \cdot m^{n-1} \cdot 1^1 + \ldots + \binom{n}{n} \cdot m^0 \cdot 1^n \\ &= (m+1)^n \end{aligned} \qquad (27)$$

$$
\begin{aligned}
CN(n+1,m) &= \binom{n+1}{0} \cdot CN(n+1,m-1) + \binom{n+1}{1} \cdot CN(n,m-1) + \ldots + \binom{n+1}{n+1} \cdot CN(0,m-1) \\
&= \binom{n+1}{0} \cdot (m-1)^{n+1} \cdot 1^0 + \binom{n+1}{1} \cdot (m-1)^n \cdot 1^1 + \ldots + \binom{n+1}{n+1} \cdot (m-1)^0 \cdot 1^{n+1} \\
&= m^{n+1}
\end{aligned}
\tag{28}
$$

Equations (27) and (28) prove that the hypothesis is true for both $CN(n+1, m)$ and $CN(n, m+1)$ so it is concluded that

$$CN(n, m) = m^n \tag{29}$$

is true for all cases where $m, n \geq 1$. Table 4.3 shows an example of all allocation result for the case with 2 customers in an overlapping area served by 4 plants.

Table 4.3: All Possible Allocations for the Set with 2 Customers into 4 Plants

No.	R_1	R_2	R_3	R_4	$[\lvert R_1\rvert, \lvert R_2\rvert, \lvert R_3\rvert, \lvert R_4\rvert]$
1	∅	∅	∅	{1,2}	[0,0,0,2]
2	∅	∅	{1}	{2}	[0,0,1,1]
3	∅	∅	{2}	{1}	
4	∅	∅	{1,2}	∅	[0,0,2,0]
5	∅	{1}	∅	{2}	[0,1,0,1]
6	∅	{2}	∅	{1}	
7	∅	{1}	{2}	∅	[0,1,1,0]
8	∅	{2}	{1}	∅	
9	∅	{1,2}	∅	∅	[0,2,0,0]
10	{1}	∅	∅	{2}	[1,0,0,1]
11	{2}	∅	∅	{1}	
12	{1}	{2}	∅	∅	[1,1,0,0]
13	{2}	{1}	∅	∅	
14	{1}	∅	{2}	∅	[1,0,1,0]
15	{2}	∅	{1}	∅	
16	{1,2}	∅	∅	∅	[2,0,0,0]

Customer Allocation When There Are Multiple Overlapping Regions So All Plants Cannot Serve All Customers in the Overlapping Regions (CL2)

The partitioning problem becomes more complex when there are more than two plants and there are customers that can be served by some of the plants but not others because of the product lifetime or time horizon. Example 4.2 serves to illustrate this situation.

Example 4.2 Consider 7 customers and 3 plants in the planned area as shown in the Figure 4.5, it has $A_{\{1\}}=\{5\}$, $A_{\{2\}}=\{6\}$, $A_{\{3\}}=\{7\}$, $A_{\{1,2\}}=\{2\}$, $A_{\{1,3\}}=\{3\}$, $A_{\{2,3\}}=\{4\}$, $A_{\{1,2,3\}}=\{1\}$, $A = \{1, 2, 3, 4, 5, 6, 7\}$.

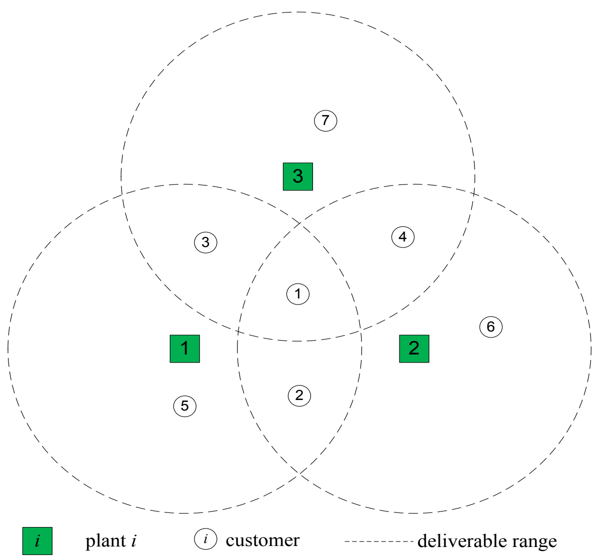

Figure 4.5: Customers and Plants Locations Graph for the Example 4.2

Figure 4.5 shows that there are 7 mutually exclusive disjoint service sets in this example: $A_{\{1\}}$, $A_{\{2\}}$, $A_{\{3\}}$, $A_{\{1,2\}}$, $A_{\{1,3\}}$, $A_{\{2,3\}}$, $A_{\{1,2,3\}}$. Each customer belongs to one disjoint service set. The customer allocation in each disjoint service set is independent to others. For example, suppose the customer 3 in $A_{\{1,3\}}$ is assigned to plant 1, this fact will not affect any other disjoint service set allocation result. So with these characteristics, Example 4.2 can be separated into 7 sub-problems. By using equation (28) in Deduction 4.2, the customer allocations number for each sub-problem are as follows: $CN(A_{\{1\}})$ =1, $CN(A_{\{2\}})$ = 1, $CN(A_{\{3\}})$ = 1, $CN(A_{\{1,2\}})$= 2, $CN(A_{\{1,3\}})$ = 2, $CN(A_{\{2,3\}})$ = 2, CN($A_{\{1,2,3\}}$) = 3. Table 4.4 shows all possible allocation result for each disjoin service set.

Table 4.4: All Possible Out Coming Result, $Out(A_\Theta)$, for All Disjoin Service Set, A_Θ, in Example 4.2

		$Out(A_\Theta)$		
A_Θ	Number	R_1	R_2	R_3
$A_{\{1\}}$	1	{5}		
$A_{\{2\}}$	2		{6}	
$A_{\{3\}}$	3			{7}
$A_{\{1,2\}}$	4	{2}	∅	
	5	∅	{2}	
$A_{\{1,3\}}$	6	{3}		∅
	7	∅		{3}
$A_{\{2,3\}}$	8		{4}	∅
	9		∅	{4}
$A_{\{1,2,3\}}$	10	{1}	∅	∅
	11	∅	{1}	∅
	12	∅	∅	{1}

Since the allocation result for each disjoint service set is independent of the others, the number of possible allocations for the set A={1,2,3,4,5,6,7} is the product of the number of allocations in each of the disjoint service sets, $CN(A_\Theta)$. In Example 4.2, $CN(A) = CN(A_{\{1\}})*CN(A_{\{2\}})*\ldots* CN(A_{\{1,2,3\}})$=1*1*1*2*2*2*3=24. Table 4.5 shows all possible customer allocations result for Example 4.2.

Table 4.5: All Customer Allocations Results for Example 4.2

No.	Assigned to plant i				R_1	R_2	R_3
	$A_{\{1,2\}}=\{2\}$	$A_{\{1,3\}}=\{3\}$	$A_{\{2,3\}}=\{4\}$	$A_{\{1,2,3\}}=\{1\}$			
1	1	1	1	1	{5,2,3,1}	{6,4}	{7}
2	1	1	1	2	{5,2,3}	{6,4,1}	{7}
3	1	1	1	3	{5,2,3}	{6,4}	{7,1}
4	1	1	2	1	{5,2,3,1}	{6}	{7,4}
5	1	1	2	2	{5,2,3}	{6,1}	{7,4}
6	1	1	2	3	{5,2,3}	{6}	{7,4,1}
7	1	2	1	1	{5,2,1}	{6,4}	{7,3}
8	1	2	1	2	{5,2}	{6,4,1}	{7,3}
9	1	2	1	3	{5,2}	{6,4}	{7,3,1}
10	1	2	2	1	{5,2,1}	{6}	{7,3,4}
11	1	2	2	2	{5,2}	{6,1}	{7,3,4}
12	1	2	2	3	{5,2}	{6}	{7,3,4,1}
13	2	1	1	1	{5,3,1}	{6,2,4}	{7}
14	2	1	1	2	{5,3}	{6,2,4,1}	{7}
15	2	1	1	3	{5,3}	{6,2,4}	{7,1}
16	2	1	2	1	{5,3,1}	{6,2}	{7,4}
17	2	1	2	2	{5,3}	{6,2,1}	{7,4}
18	2	1	2	3	{5,3}	{6,2}	{7,4,1}
19	2	2	1	1	{5,1}	{6,2,4}	{7,3}
20	2	2	1	2	{5}	{6,2,4,1}	{7,3}
21	2	2	1	3	{5}	{6,2,4}	{7,3,1}
22	2	2	2	1	{5,1}	{6,2}	{7,3,4}
23	2	2	2	2	{5}	{6,2,1}	{7,3,4}
24	2	2	2	3	{5}	{6,2}	{7,3,4,1}

In general, all customers can be assigned into one of N disjoint service sets so that $A = A_{\Theta 1} \cup \quad \cup \quad \cup$. The number of possible allocations in each disjoint service set, $CN(A_{\Theta i})$, can be calculated with formula (28) shown in CL1. Since these allocations are independent, the total number of possible allocations for set A is to the product of all $CN(A_{\Theta i})$ from the disjoin service sets:

$$CN(A) = \prod_{i=1}^{N} CN(A_{\Theta i}) \tag{30}$$

Recall from equations (25) and (29), $CN(n, m) = CN(A_\Theta) = m^n$, where $|A_\Theta|=n$ and $|\Theta|=m$. For each overlapping region, m ≥ 2, the number of allocation result will be greater than or equal to 2^n so $CN(A_\Theta) \geq 2^n$ when $|\Theta|\geq 2$. Hence, if there are M customers in the overlapping region, the total number of possible partition will great than or equal to 2^M. This means that allocating customers to different plants cannot be solved within polynomial time and this problem is also NP hard. It is not possible to use enumeration when the total number of customers that can be served by multiple plants is large so attention is now turned to presenting several heuristics for generating the partition that use two different strategies: (I) fixed allocation, and (II) random allocation.

Heuristic customer allocation methods

In the previous section, it was shown that allocating customers in the overlapping region to different plants is an NP hard problem; hence, the use of heuristic methods to make the assignments is justified and necessary for finding approximate solutions to larger problems. In this section, two different assignment strategies are proposed: (I) fixed allocation and (II) random allocation. Fixed allocation methods make the final assignment of customers to plants using only the basic information regarding customers and plants like location, demands and capacities. They always generate the same result as long as the basic information is unchanged. Recall the hierarchy of decisions in the MPIPDSPPP in Figure 4.3. This is an example of fixed allocation and the customer allocation from this approach will always be the same as long as the basic information remains unchanged. Random allocation methods, on the other hand, use the basic information in fixed approaches but also introduce randomness in the process in the spirit of heuristic procedures like evolutionary algorithms. They can provide different customer allocation results even when the basic information remains the same. The following sections provide details about several specific approaches within each of these general types of methods.

Fixed Allocation Methods

Fixed allocation methods utilize basic information about the problem like customer and plant locations as well as demands, to make the assignment of customers to plants. As such, applying them to a situation multiple times will produce the same results unless the basic information changes. The advantages of fixed allocation methods are that they can typically be applied very quickly and the results are reproducible. The weakness is that these methods are typically designed based on data from problems with certain features; however, when they are applied to situations in which these features are different, they can produce solutions that are not very good. This should not be construed as implying that fixed allocation methods are not useful because they can certainly be excellent choices in practice when applied to problems similar to the ones from which they were designed; however, their limitations, especially those related to domain of applicability, need to be carefully noted. Three different fixed allocation methods are now proposed.

Method 1: Distance Priority Method (M1)

The first method exploits a common sense approach by assigning each customer in the overlapping region to the nearest plant. In other words, customers will be assigned to the plant that requires the minimum variable travel cost for direct delivery. This method will obviously perform well when a large proportion of the deliveries are direct from plant to customer but might be less useful when customer locations are such that multiple stops on each trip are best. Example 4.3 provides an instance for the distance priority method.

Note: For calculation convenience, the measurement unit for time and distance are unit time and unit distance. Also, it is assumed that the vehicle speed is constant = 1 unit distance / unit time.

Example 4.3: Consider the case with two plants and three customers. The demand for the customers are q_1=4, q_2=4, q_3=8. The product has a constant lifetime B=25 and the time horizon for each period H = 50. The production rate for the plants are r_1=1, r_2=2. There are two types of vehicles available for the delivery, with the capacity, fixed setup cost and unit distance cost are C_1=8, F_1=100, R_1=1, C_2=10, F_2=110, R_2=1.1. The location of the customers and plants are shown in Figure 4.6 as are the travelling distance between each customer and each plant.

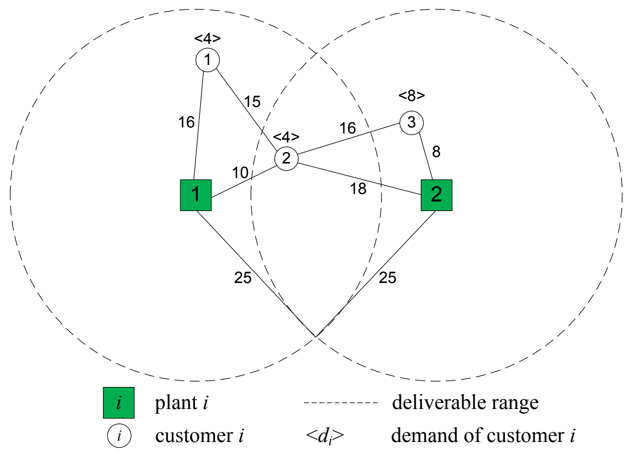

Figure 4.6: Customers and Plants Locations Graph for the Example 4.3

Applying the distance priority method, customer 2 is assigned to the plant 1 because plant 1 is 10 units from the customer while plant 2 is 18 units. By enumerating all possible feasible solutions with this customer allocation result, the optimal solution is found to be one type 1 vehicle for plant 1 and one type 1 vehicle for plant 2 with the production lot size and the detailed schedule for production and distribution shown in Figure 4.7. In this figure, j = customer j, Pi = plant i, and d_j = time to produce customer j's demand. The best schedule generated by Method 1 (M1) for Example 3.3 is shown in Figure 4.7. The system total transportation cost with the schedule is:

Total cost = Fixed cost for 2 type 1 vehicles + variable cost for plant 1 deliveries + variable cost for plant 2 deliveries

= 100 * 2 + 41*1 + 16* 1 = 257

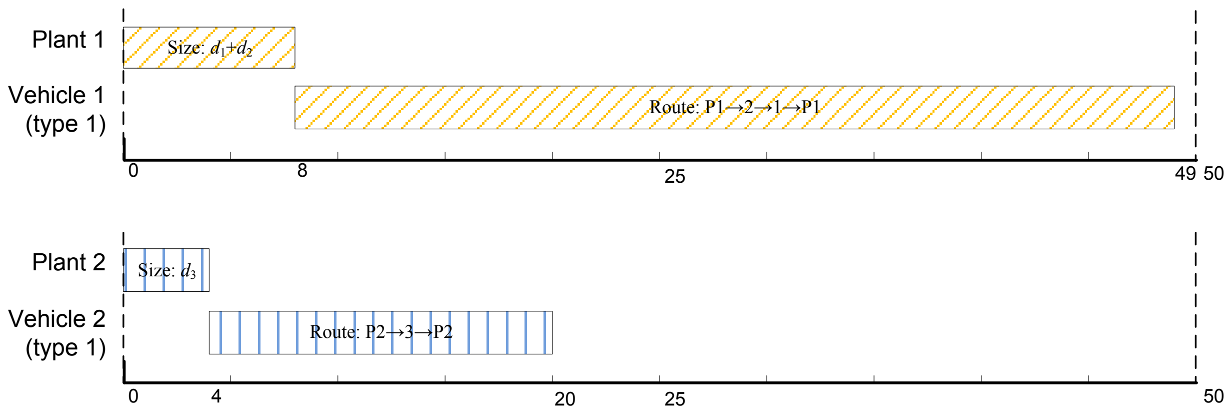

Figure 4.7: Best Solution for Example 4.3 When Allocating Customer 2 to Plant 1

Method 2: Multi-Stop Delivery Priority Method (M2)

The distance priority method is a simple method to implement as illustrated in Example 4.3; however, it is intuitive that this simplicity has a price which consists of some important limitations. For one, the distance priority method only considers the influence of direct delivery distance from each plant and ignores other important possibilities like multi-stop delivery. Obviously, delivering to multiple customers in one trip can reduce the variable cost of transportation compared to direct delivery in certain situations. Including this concept is the basis for the multi-stop delivery priority method. This heuristic adds the ability to assign customers to the plant where there is an increased possibility that they can be included in a more cost effective multi-stop delivery trip rather than direct delivery. A second limitation is that the distance priority method focuses exclusively on the variable cost associated with distance and ignores the special constraints of the problem such as the product lifetime constraint and vehicle capacity constraint.

This heuristic method includes both problem constraints and consideration of the possibility for a customer to be added to a multi-stop delivery trip. Define a "check customer" as a customer that can be served by multiple plants, a "check plant" is a plant can serve the check customer, a "feasible check trip" is a trip that starts its route at a check plant, delivers to $n \geq 2$ customers which includes check customers, and returns to the check plant while satisfying lifetime constraint and capacity constraint. By definition, each check customer *j* can be served by more than one check plant. In this heuristic, the assignment decision is based on the number of feasible check trips that each check plant can provide. Specifically, customer *j* is assigned to the check plant that can serve it that has the highest number of feasible check trips. To find the number of feasible check trips, the following four-step process is used:

(1) Find an arbitrary subset Q of $A_{\{p\}}$ which not include check customer j, i.e. $Q \subset A_{\{p\}}$ and $j \notin Q$. Recall, the $A_{\{p\}}$ is the servable customer set for plant p

(2) Add check customer j to the subset Q to be a new set Q'

(3) Check the capacity constraint to delivery all customer in Q' with one multi-stop delivery. Record number of vehicles type can satisfy the capacity constraint as s, if s=0, then go to step 5, otherwise go to step 4

(4) Check the lifetime constraint to delivery all customers in Q' with one multi-stop delivery. If there are n feasible trips, the feasible check trip number increases by $n*s$.

(5) Repeat the step (1) to (3) until all possible subsets are tested.

This process is short but the computations are complex. As shown in Kamke (1950), the possible number of subsets is 2^n when there are n elements in the set. To enumerate all possible subsets has a complexity of at least $O(2^n)$. Also, to find a possible trip in step 3 is similar to the TSP problem which has the complexity near $O(n!)$.

To simplify this process, this research only considers the multi-stop delivery with 2 total customers which means that the heuristic only needs to consider all subsets with 1 customer in $A_{\{p\}}$ in step 1. This assumption reduces the complexity of step 1 from $O(2^n)$ into $O(n)$. Also, the complexity of step 3 is reduced from $O(n!)$ to $O(1)$. We now refine the idea of check trip by redefining a "pair trip", $T_{i,j,p,s}$, to be the route that starts at plant p, visits customers i and check customer j, and then returns to the plant p using a vehicle of type s. To check the feasibility of a pair trip, the heuristic only need to check 2 possible trips routes against the product lifetime constraint; namely, the heuristic must seek evaluate $\tau_{pi} + \tau_{ij} \leq B$ and $\tau_{pj} + \tau_{ij} \leq B$, and the vehicle capacity constraint which is $q_i + q_j \leq C_s$. With the simplification and constraint checks, check customer j is assigned to the plant that can provide the greatest number of feasible pair trips. If there is a tie in the number of trips between multiple plants, the distance priority method will be used as the tie-breaking criteria. In Example 4.3, the heuristic must allocate check customer 2 to a plant. Plant 1 has 2 pair trips, $T_{2,1,1,1}$ and $T_{2,1,1,2}$ because check customer 2 can be delivered in the same route P1→2→1→P1 with customer 1 and both vehicle types 1 and 2 for this route, the other route P1→1→2→P1 violates the lifetime constraint. Plant 2 has no feasible pair trips because neither vehicle type has sufficient capacity to deliver a multi-stop route consisting of customers 2 and 3. Customer 2 is therefore assigned to plant 1.

The pseudo code in Figure 4.8 is used to allocate a check customer k to a check plant p. where $k \in A_{\Theta}$, $p \in \Theta$. Recall Θ is a set of plant. A_{Θ} is a set of customers that can be and only can be served by the plants in the set Θ.

```
While (not done)
    Select a plant p from set Θ.
    if (|Θ| = 1)
        allocate customer k to plant p. Finished.
    else
        for each ( customer i in A{p}, vehicle type s in S )
            if ( trip T_{k,i,p,s} is feasible)
                count[p] = count[p] + 1
        end for
    if all plant in Θ had been selected
        done = true
    else
        done = false
end while
allocate customer k to the plant give the highest count, use distance priority
method when there is a tie.
```

Figure 4.8: Pseudo Code for the Calculation of Number of Feasible Trips

Compared to the distance priority method, the multi-stop delivery priority tries to reduce the system cost by allocating check customers to the plant which can provide them with the greatest chance of being included in multi-stop delivery. It is more sensitive to the relative distance between customers and relative customer demands. Example 4.4 illustrates some of the advantages of the multi-stop delivery priority method.

Example 4.4: Consider the same basic problem described Example 4.3 except that customer demand has been changed. The demands for customers are $q_1 = 8$, $q_2 = 4$, $q_3 = 4$. The remainder of the problem is the same: the product has a constant lifetime $B = 25$, the time horizon for each period $H = 50$, the production rate for the plants are $r_1 = 1$ and $r_2 = 2$, there are two types of vehicles available for the delivery with capacity, fixed setup cost and unit distance cost of $C_1 = 8$, $F_1 = 100$, $R_1 = 1$ and $C_2 = 10$, $F_2 = 110$, $R_2 = 1.1$. The location of the customers and plants are shown in Figure 4.9 with the distance between each customer and each plant also shown in the figure.

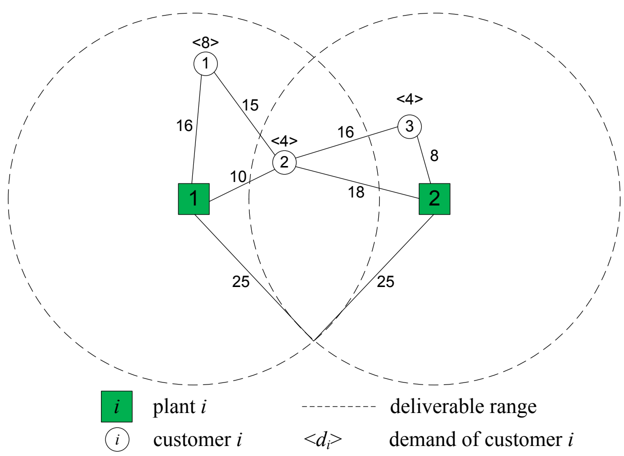

Figure 4.9: Customers and Plants Locations Graph for the Example 4.4

Comparison Solution 1 for Example 4.4: Solving Example 4.4 with the Distance Priority Method (M1)

Example 4.4 is now solved with the Method 1. The only Type III customer that can be served by both plants is customer 2 and the distance priority method assigns it to plant 1 because the direct delivery cost for customer 2 via direct delivery is minimum with plant 2. This assignment, however, does not lead to the minimum system transportation cost because assigning customer 2 to plant 2 allows customers 2 and 3 to be delivered in one trip which is less costly that the three direct deliveries that are required when customer 2 is assigned to plant 1. That is, it is not possible to combine the trips of customers 1 and 2 because their total demand exceeds the capacity of the largest vehicle so the best schedule for Example 4.4 using Method 1 (M1) (i.e. allocate customer 2 to plant 1) is shown in Figure 4.10. The system total transportation cost with the schedule is:

Total cost = Fixed cost for 3 type 1 vehicles + variable cost for plant 1 deliveries (trip 1 and 3) + variable cost for plant 2 deliveries (trip 7)

= 100*3 + 32*1 + 20*1 + 16*1= 368

The best schedule was guaranteed by solving 2 IPDSPPP problems with MIP model using AMPL + GUROBI.

Table 4.6: All Feasible Trips for Example 4.4

No.	Start Plant	Vehicle Type	Trip Route	Trip Variable Cost
1	1	1	P1→C1→P1	32
2	1	2	P1→C1→P1	35.2
3	1	1	P1→C2→P1	20
4	1	2	P1→C2→P1	22
5	2	1	P2→C2→P2	36
6	2	2	P2→C2→P2	39.6
7	2	1	P2→C3→P2	16
8	2	2	P2→C3→P2	17.6
9	2	1	P2→C3→C2→P2	42
10	2	2	P2→C3→C2→P2	46.2

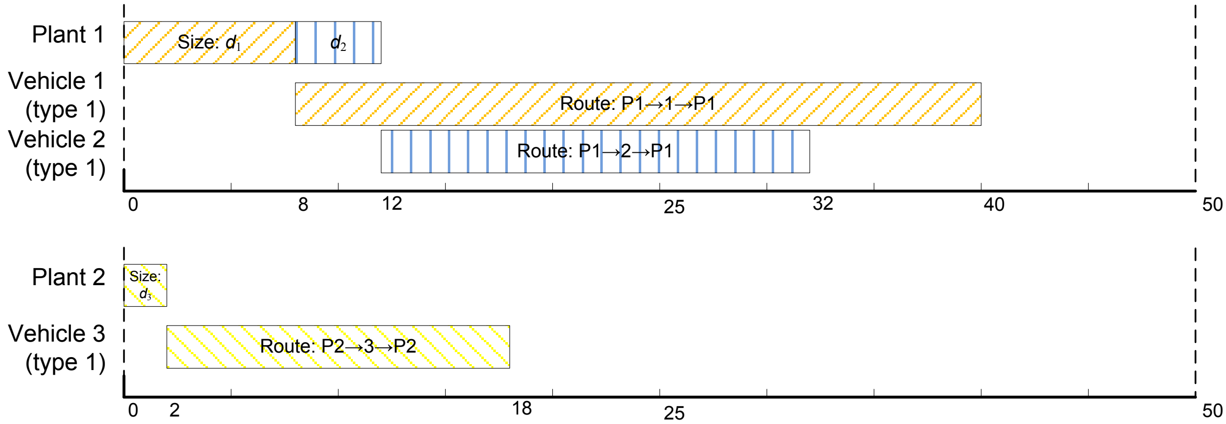

Figure 4.10: Best Schedule for Example 4.4 with Method 1

Comparison Solution 2 for Example 4.4: Solving Example 4.4 with Multi-Stop Delivery Priority Method (M2)

To solve Example 4.4 using Method 2, the first requirement is to determine the check customers; in this case it is only customer 2. Then, for each plant *i*, it is necessary to count the total number of feasible pair trips that include customer 2. As shown in the comparison solution 1 for Example 4.4, it is not feasible to deliver both customer 1 and customer 2 in one delivery due to vehicle capacity restrictions so there are no feasible pair trips emanating from plant 1. For plant 2, there are 2 feasible pair trips: trip 9 and trip 10 as seen in Table 4.6. The rule is then applied that allocates customer 2 to the plant with the most feasible pair trips which, in this case, is plant 2. The best schedule, then, is shown in Figure 4.11 and the system total transportation cost with the schedule is:

Total cost = Fixed cost for 2 type 1 vehicles + variable cost for plant 1 deliveries (trip 1) + variable cost for plant 2 deliveries (trip 9)

$$= 100*2 + 32*1 + 42*1 = 274$$

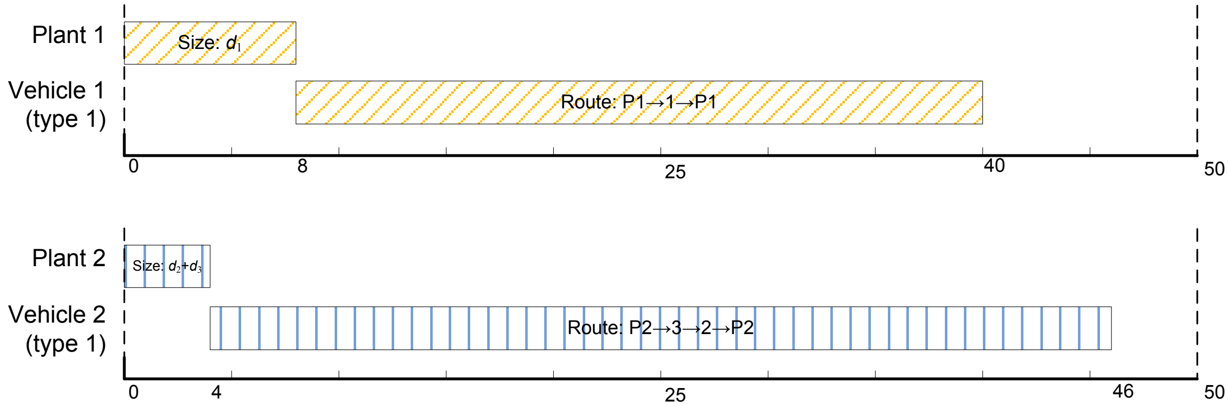

Figure 4.11: Best Schedule for Example 4.4 with Method 2\

In Example 4.4, the total cost for Method 2 is much less than the one obtained in Method 1. It saves the money by using the multi-stop delivery rather than direct delivery.

Method 3: Minimal Multi-Stop Delivery Cost Priority Method (M3)

Similar to Method 2, the minimal multi-stop delivery cost priority method also assigns customers to the plant where they have the greatest opportunity of being included in a multi-stop delivery trip to reduce transportation cost; however, this method makes the decision based on the minimum cost of the feasible pair trip variable transportation cost each plant can provide rather than including the total number of feasible pair trips from each plant. Since the objective is to reduce the total system transportation cost, allocating check customers to plants that provide the lower transportation cost seems a very natural way to partition. Method 3, then, allocates the selected check customer to the plant that provides the check customer with the minimum variable cost feasible pair trip. Example 4.5 illustrates the minimum delivery cost priority method.

Example 4.5: Consider a problem with the same basic structure as Example 4.3 but now with different customer demands and locations. The demands of the three customers are $q_1 = 5$, $q_2 = 4$, $q_3 = 4$. The product has a constant lifetime $B = 25$ and the time horizon for each period H = 50. The production rates for the plants are $r_1 = 1$, $r_2 = 2$. There are two types of vehicles available for the delivery, with capacity, fixed setup cost and unit distance cost equal to $C_1 = 8$, $F_1 = 100$, $R_1 = 1$, $C_2 = 10$, $F_2 = 110$, $R_2 = 1.1$. The locations of the customers and plants are shown in Figure 4.12, the travelling distance between each customer and each plant are also shown in the figure.

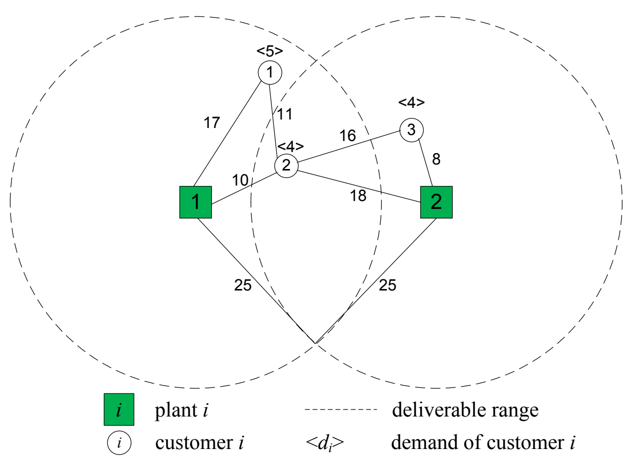

Figure 4.12: Customers and Plants Locations Graph for the Example 4.4

Comparison Solution 1 for Example 4.5: Solve Example 4.5 with Method 2, Multi-Stop Delivery Priority Method (M2)

To provide a comparison later, this example is first solved using the using the minimal multi-stop delivery priority method which is the strategy in which each check customer is assigned to the plant that provides the greatest number of potential pair trips. The logic is that this will maximize the chances that the customer can become part of a pair trip so delivery will cost less than direct delivery. Table 4.7 enumerates all feasible trips for Example 4.5. From this table it can be seen that for plant 1, there is 1 pair trip, trip 5, and for plant 2 there are 2 pair trips, trips 10 and 11. Using the multi-stop delivery priority method, customer 2 will be allocated to plant 2 because 2 pair trips is more than 1. The best schedule generated by Method 2 (M2) is shown in Figure 4.13 and the system total transportation cost is

Total cost = Fixed cost for 2 type 1 vehicles + variable cost for plant 1 deliveries (trip 1) + variable cost for plant 2 deliveries (trip 10)

= 100*2 + 34*1 + 42*1 = 276.

Table 4.7: All Feasible Trips for Example 4.5

No.	Start Plant	Vehicle Type	Trip Route	Trip Variable Cost
1	1	1	P1→C1→P1	34
2	1	2	P1→C1→P1	37.4
3	1	1	P1→C2→P1	20
4	1	2	P1→C2→P1	22
5	1	2	P1→C2→C1→P1	41.8
6	2	1	P2→C2→P2	36
7	2	2	P2→C2→P2	39.6
8	2	1	P2→C3→P2	16
9	2	2	P2→C3→P2	17.6
10	2	1	P2→C3→C2→P2	42
11	2	2	P2→C3→C2→P2	46.2

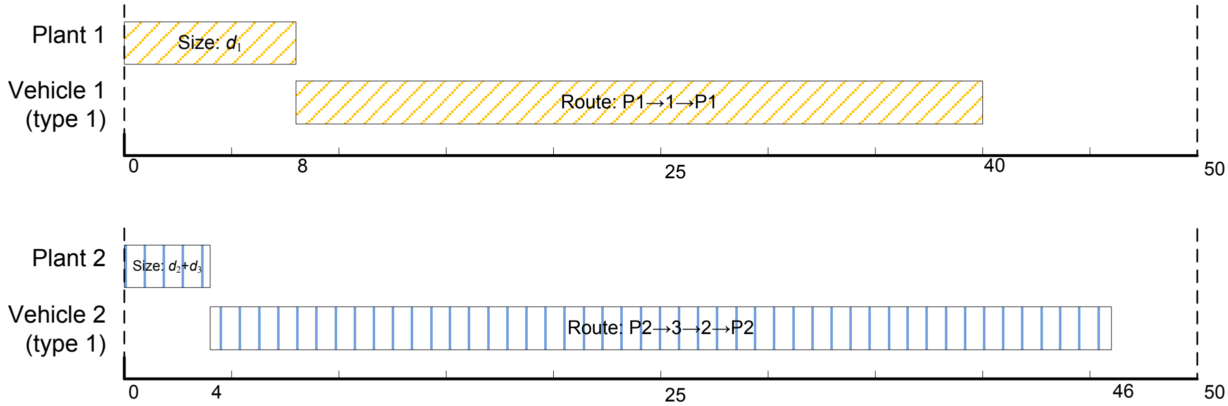

Figure 4.13: Best Schedule for Example 4.5 with Method 2

Comparison Solution 2 for Example 4.5: Solve Example 4.5 with Minimal Multi-Stop Delivery Cost Priority Method (M3)

This example is now solved using the minimal multi-stop delivery priority method. Recall that this method allocates check customers to plant that provides the feasible pair trip with minimum variable transportation cost. As shown in Table 4.7, there are 3 pair trips for check customer 2: 1) trip 5 which has a variable cost of 41.8, 2) trip 10 with a cost of 42, and 3) trip 11 with a cost of 46.2. As such, check customer 2 is assigned to plant 1 because it that is the plant associated with trip 5 which has the minimum variable transportation cost of all feasible pair trips. Note that the minimal multi-stop delivery cost priority method makes a different assignment from the multi-stop delivery priority method. When the number of pair trips is considered, check customer 2 is allocated to plant 2 whereas the minimum variable cost allocation is to the plant 1. The best schedule generated by Method 3 is shown in Figure 4.14 and the system total transportation cost is

Total cost = fixed cost for 1 type 1 vehicles + fixed cost for 1 type 2 vehicles

+ variable cost for plant 1 deliveries (trip 5) + variable cost for plant 2 deliveries (trip 8)

= 100*1 + 110*1 + 41.8*1 + 16*1 = 267.8

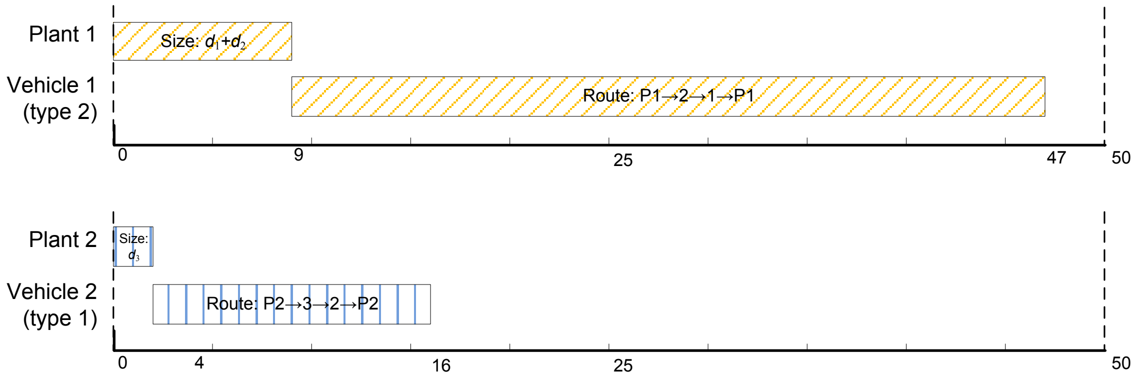

Figure 4.14: Best Schedule for Example 4.5 with Method 3

In Example 4.5, the total cost for Method 3 is less than the total cost for Method 2. Furthermore, even though both methods use 2 vehicles for delivery, the schedule generated by Method 3 has much more idle time for vehicle 2 than the schedule generated by Method 2. This idle time allowed the plant 2 to get more trips without adding extra vehicles.

Method 4: Average Multi-Stop Delivery Cost Priority Method (M4)

As might be anticipated, Method 4 extends the idea of Method 3 by considering the average cost of all feasible pair trips rather than the minimum pair trips cost. This is motivated by the very real possibility that the vehicle paired with the check customer to produce minimum variable cost might not be available because it is paired with another customer for greater saving. It is also possible that because multiple customers are allowed on a single trip, a larger variable trip cost could lead to more consolidation and the delivery fleet size could be smaller. Sometimes the savings of a smaller vehicle fleet size is of more beneficial than the variable costs of multiple larger vehicles. Also, this method does not restrict trips to include only 2 customers because adding a check customer to a route that eventually includes more than 2 customers could produce a better solution. This method, then, combines ideas in Methods 2 and 3 in that both all of the feasible pair trips are included as is the variable cost for each with the result of assigning check customers based on the average cost.

Example 4.6: This example considers 5 customers and 2 plants in the system. The demands for the customers are q_1 = 3, q_2 =3, q_3 = 6, q_4 =1, q_5 = 1. The product has a constant lifetime B = 25 and the time horizon for each period H = 50. The production rates for the plants are r_1= 1, r_2=1. There are two types of vehicles available for delivery, with the capacity, fixed setup cost and unit distance cost equal to C_1 = 8, F_1 = 100, R_1 = 1, C_2 = 12, F_2 =110, R_2 = 1.1. The location of the customers and plants are shown in Figure 4.15; the travelling distance between each customer and each plant is shown in the figure.

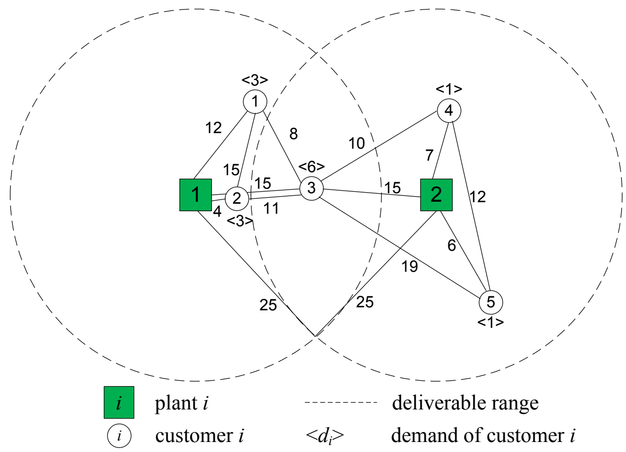

Figure 4.15: Customers and Plants Locations Graph for the Example 4.6

Comparison Solution 1 for Example 4.6: Solving Example 4.6 with Multi-Stop Delivery Priority Method (M2)

As before, several methods will be used to solve the problem and the solution results will be compared later. First method is using the multi-stop delivery priority method. As such, each check customer will be assigned to the plant that has the maximum number of feasible pair trips to offer. Table 4.8 provides all feasible trips for the Example 4.6. For plant 1, it could find 4 feasible pair trips: trip 7, 8, 9 and 10 (trip 11, 12 and 13 are not pair trips). For plant 2, it could find 6 feasible pair trips: trip 20, 21, 22, 23, 24 and 25 (trip 26, 27, 28 and 29 are not pair trips). So the customer 3 will assign to plant 2. The best schedule generated by Method 2 is shown in Figure 4.16 and the total system total transportation cost is

Total cost = Fixed cost for 2 type 1 vehicles + variable cost for plant 1 deliveries (trip 11) + variable cost for plant 2 deliveries (trip 18 and 22)

= 100*2 + 31*1 + 32*1 + 12*1 = 275

Table 4.8: All Feasible Trips for Example 4.6

No.	Start Plant	Vehicle Type	Trip Route	Trip Variable Cost	Pair Trip
1	1	1	P1→C1→P1	24	
2	1	2	P1→C1→P1	26.4	
3	1	1	P1→C2→P1	8	
4	1	2	P1→C2→P1	8.8	
5	1	1	P1→C3→P1	30	
6	1	2	P1→C3→P1	33	
7	1	2	P1→C1→C3→P1	38.5	Y
8	1	2	P1→C3→C1→P1	38.5	Y
9	1	2	P1→C2→C3→P1	33	Y
10	1	2	P1→C3→C2→P1	33	Y
11	1	1	P1→C2→C1→P1	31	
12	1	2	P1→C2→C1→P1	34.1	
13	1	2	P1→C2→C3→C1→P1	38.5	
14	2	1	P2→C3→P2	30	
15	2	2	P2→C3→P2	33	
16	2	1	P2→C4→P2	14	
17	2	2	P2→C4→P2	15.4	
18	2	1	P2→C5→P2	12	
19	2	2	P2→C5→P2	13.2	
20	2	1	P2→C3→C4→P2	32	Y
21	2	2	P2→C3→C4→P2	35.2	Y
22	2	1	P2→C4→C3→P2	32	Y
23	2	2	P2→C4→C3→P2	35.2	Y
24	2	1	P2→C5→C3→P2	40	Y
25	2	2	P2→C5→C3→P2	44	Y
26	2	1	P2→C4→C5→P2	25	
27	2	2	P2→C4→C5→P2	27.5	
28	2	1	P2→C5→C4→P2	25	
29	2	2	P2→C5→C4→P2	27.5	

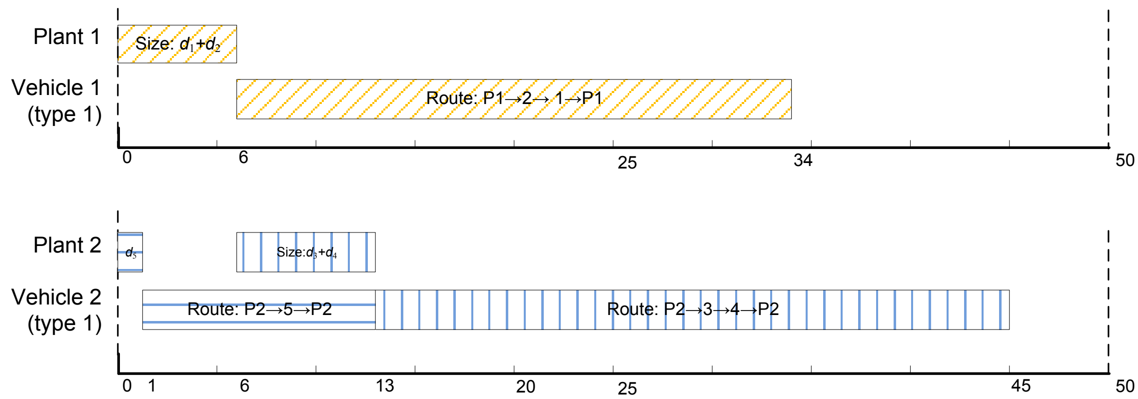

Figure 4.16: Best Schedule for Example 4.6 with Method 2

Comparison solution 2 for Example 4.6: Solving Example 4.6 with Minimal Multi-Stop Delivery Cost Priority Method (M3)

The second compared method is the Minimal Multi-Stop Delivery Cost Priority method. As such, each check customer will be assigned to the plant that provides the feasible pair trip with minimum variable transportation cost. As shown in Table 4.8, there are 10 feasible pair trips (trip 7, 8, 9, 10, 20, 21, 22, 23, 24, 25). The trip 20 and 22 which both provided by plant 2 give the minimum variable transportation cost equal to 32. As such, check customer 2 is assigned to plant 1. Note that the minimal multi-stop delivery cost priority method makes the same allocation with the multi-stop delivery priority method. So, for Example 4.6, minimal multi-stop delivery cost priority method and the multi-stop delivery priority method lead to the same best solution as shown in Figure 4.16. The calculation of total transportation cost is also shown in comparison solution 1 for Example 4.6.

Comparison solution 3 for Example 4.6: Solve Example 4.6 with Average Multi-Stop Delivery Cost Priority Method

The average multi-stop delivery cost priority method considers both feasible pare trip number and its relative transportation cost. For each check customer, it calculates the average feasible pare trip cost for each plant, and allocate the check customer to the one which provides the minimum average feasible pare trip cost. As mentioned in comparison solution 1, the plant 1 has 4 feasible pare trips: trip 7, 8, 9 and 10 with average transportation cost = (38.5 + 38.5 + 33 + 33) / 4 = 35.75. The plant 2 has 6 feasible pare trips: trip 20, 21, 22, 23, 24 and 25 with average transportation cost = (32 + 35.2 + 32 + 35.2 + 40 + 44) / 6 = 36.4. So the customer 2 will assign to the plant 1. Note that the Average Multi-Stop Delivery Cost Priority method makes a different assignment with previous 2 methods. The best schedule generated by Method 4 is shown in Figure 4.17, and the system total transportation cost is

Total cost = fixed cost for 1 type 1 vehicles + fixed cost for 1 type 2 vehicles
+ variable cost for plant 1 deliveries (trip 13) + variable cost for plant 2 deliveries (trip 26)
= 110*1 + 100*1 + 38.5*1 + 25*1 = 273.5

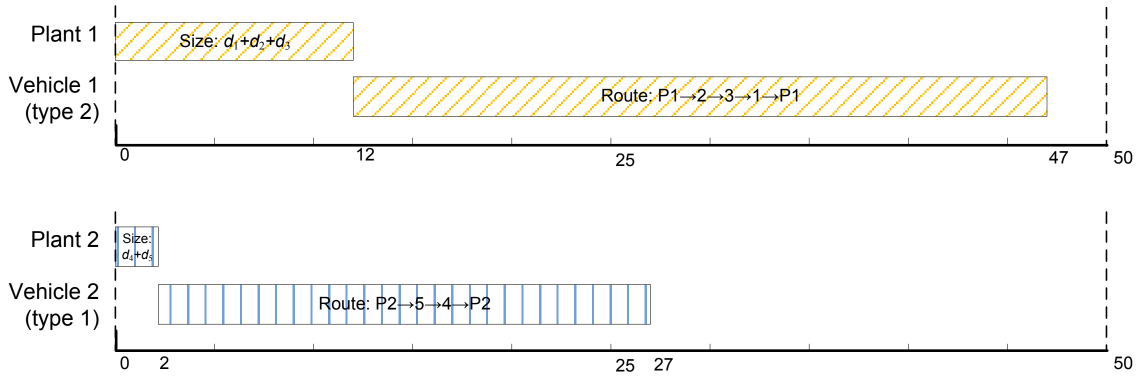

Figure 4.17: Best Schedule for Example 4.6 with Method 4

In Example 4.6, 2 best schedules are generated by 3 different methods (Method 2 and 3 generate the same best schedule), the best schedule generate by Method 4 has lower transportation cost than the other ones. Furthermore, Method 4 also generates a more compressed transportation schedule for vehicle 2 and saves the vehicle 2 lots of idle time.

Random Allocation Methods

Random allocation refers to methods that use both the basic information in fixed allocation methods and includes randomness in the assignment scheme. The motivation for using random allocation is to generate more than one customer allocation result from one method for each problem. One result is that that these methods can produce different check customer assignments for the same problem.

Method 5: Equal Randomness Method (M5)

The first way that randomness is included in the assignment heuristic is the equal randomness method. Here, check customer *j* has an equal probability of being assigned to any of the *n* plants that can serve it. The advantage of this method is that it is simple and easy to implement. The weakness is that improvement using any insights based on the basic information is traded for a pure random strategy which does not intuitively seem to be a good tradeoff. Regardless, it is proposed as something of a base case for random allocation.

Method 6: Balanced Productivity Method (M6)

The equal randomness method provides each check customer with equal opportunity to be assigned to any feasible plant. One problem here is that the productivity of the plant is not considered so a check customer could be assigned to a plant where a feasible solution is not possible. Consider a system with 2 plants in which Plant 1 has a very high productivity and Plant 2 has low productivity. Assigning a check customer to Plant 1 could create a situation where the product for the check customer cannot be produced so delivery meets the constraints while Plant 2 has a plenty of capacity to server the check customer within the constraints. The balanced productivity method assigns check customers to plants based on probabilities that are tied to productivity so that plants with the lowest productivity, alternatively the highest idle time, have the greatest chance of being assigned a check customer. To implement this strategy, all check customers are randomly put in a list, suppose the first check customer in the list is customer *j* which has

n servable plants. Define the current utilization for servable plant i as, $u_i = \frac{\sum_{j \in M_i} q_j / r_i}{H}$,

where M_i={all customers already assigned to plant i}. The idle time proportion for servable plant i is, $w_i = 1 - u_i$. So, each servable plant i has the change to be assigned check customer j with weight w_i. By standard process, the weight can be changed to the probability $p_i = \frac{w_i}{\sum_{i=1}^{n} w_i}$. After the assignment of customer j finished, the utilization of the plant will be updated and the next check customer can start the assignment again. Example 4.7 illustrates the balanced productivity method.

Example 4.7: There are 5 customers 2 plants in the system. The demand for the customers are q_1 = 4, q_2 =2, q_3 = 6, q_4 =8, q_5 = 2. The product has a constant lifetime B = 25 and the time horizon for each period H = 50. The production rate for the plants are r_1= 1, r_2=2. There are two types of vehicles available for the delivery, with the capacity, fixed setup cost and unit distance cost are C_1 = 8, F_1 = 100, R_1 = 1, C_2 = 12, F_2 =110, R_2 = 1.1. The location of the customers and plants are shown in Figure 4.18, the travelling distance between each customer and each plant are also shown in the figure.

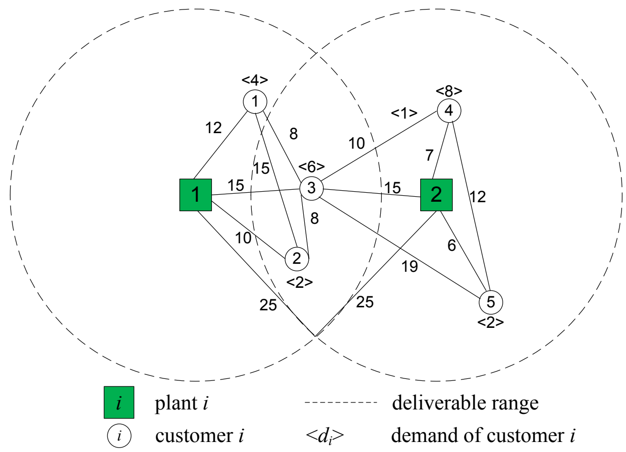

Figure 4.18: Customers and Plants Locations Graph for the Example 4.7

As shown in the Figure 4.18, there are only 2 check customers: customers 2 and 3. Customer 1 can only be allocated to Plant 1 and customer 4, 5 can only be assigned to Plant 2. The idle time proportion for plant 1 is $w_1 = 1 - \frac{4/1}{50} = 0.92$ and for plant 2 it is $w_2 = 1 - \frac{(8+2)/2}{50} = 0.9$. Suppose a random list for check customer is <3, 2>. Then the customer 3 will be assign to plant 1 with probability $p_1 = \frac{0.92}{0.92+0.9} \approx 0.51$, to plant 2 with probability $p_1 = \frac{0.9}{0.92+0.9} \approx 0.49$. Assume the customer 3 is assigned to plant 1. Then the plant 2's idle time proportion will be updated, $w_1 = 1 - \frac{(4+6)/1}{50} = 0.8$ and $w_2 = 1 - \frac{(8+2)/2}{50} = 0.9$. Check customer 2 will be assigned to plant 1 with probability $p_1 = \frac{0.8}{0.8+0.9} \approx 0.47$, to plant 2 with probability $p_2 = \frac{0.9}{0.8+0.9} \approx 0.53$.

Method 7: Simulated Annealing Method (SA)

The two previous random allocation methods can generate different customer allocation result for each application of the heuristic but both methods start with the original problem. That is, neither has any type of learning involved so that sequential applications results in progressively improved solutions. The brief outline of the simulated annealing search method (SA) previously provided in Chapter 3 noted that it is a probability based search technique that seeks the global optimum in a solution space. While it is similar to local search in the sense that is moves the solution towards better solution, SA is also different because it uses a transition probability function to accept the movement to a solution that degrades the objective function value. The motivation behind this is to avoid trapping the algorithm at local optima. Figure 4.19 presents the flow chart of the SA method for MPIPDSPPP problem.

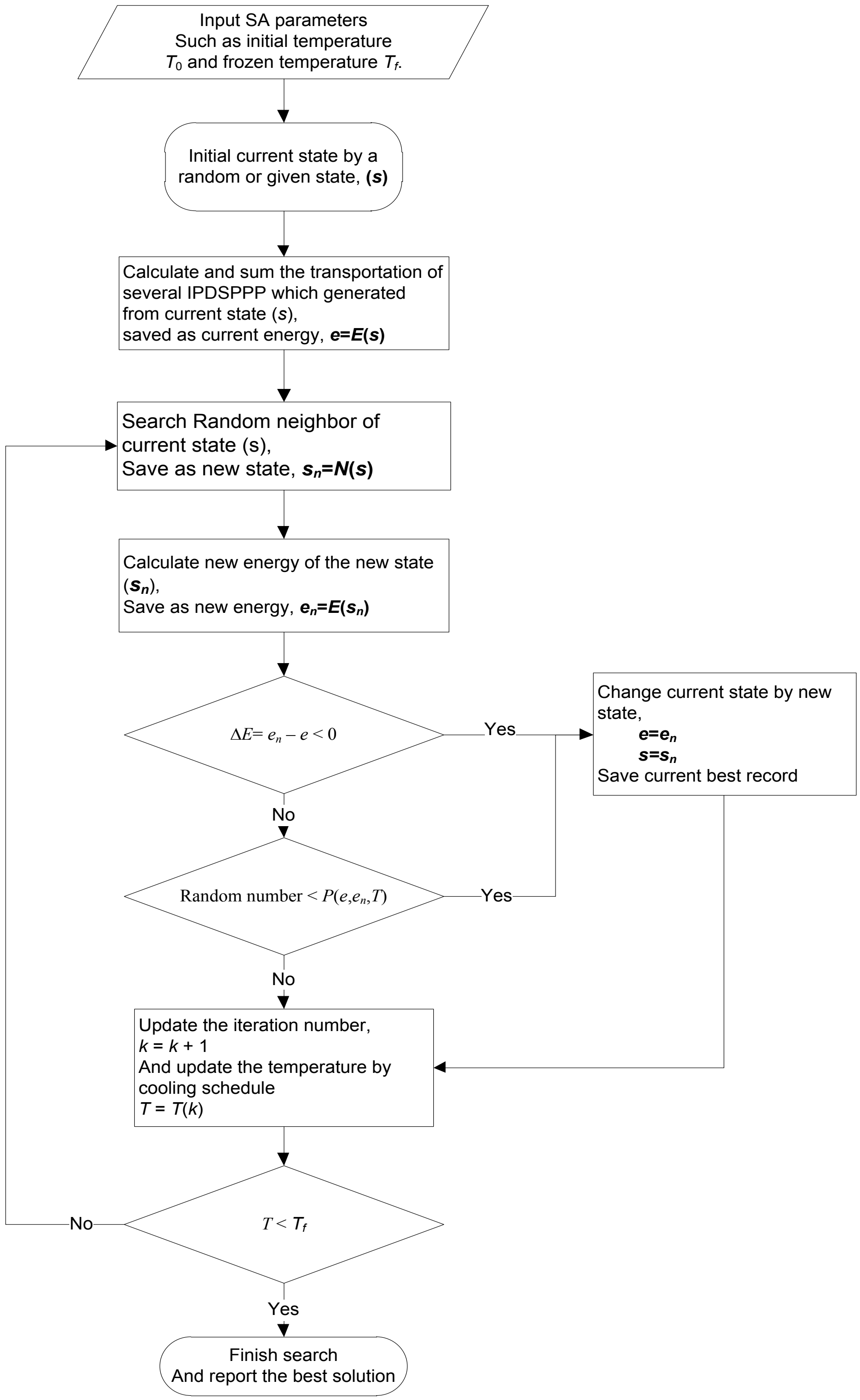

Figure 4.19: Flow Chart of the SA Method for MPIPDSPPP Problem

There are three important elements to a SA search algorithm: 1) a method to encode the current state. 2) The neighborhood function to search for neighbor states, and 3) a way to calculate the objective function value from state. The rest of the parameters required of the algorithm such as the initial temperature T and frozen temperature T_f are set once and remain static until the algorithm is restarted. Constructions of these three parts are illustrated by Example 4.8.

Example 4.8: Consider a two plant 8 customers MPIPDSPPP problem. $A_{\{1\}}= \{1, 2, 3\}$, $A_{\{2\}}=\{4, 5\}$, $A_{\{1,2\}}=\{6, 7, 8\}$.

First of all, we need to encode the solution of the assignment of customers to plants. Different with the IPDSPPP problem, this must include both identification of the customer and the plant. To achieve this, a random production permutation to represent the customer is generated as in the IPDSPPP but here it is augmented with an ally permutation that represents the plant to which each customer is assigned. The two permutations have the same dimension so that the check customer in the i^{th} location of the production permutation is assigned to the i^{th} plant in the ally permutation. Assume; for example, consider the encoding sequence in Figure 4.20. This means that check customers 6 and 7 are allocated to plant 1, and check customer 8 is allocated to plant 2.

ally permutation	2	2	1	2	1	1	1	1
production permutation	4	5	3	8	2	7	6	1

Figure 4.20: Encoding Representation of Customer Allocation

At this stage the production permutation is separate into two production permutations based on the ally permutation, one for Plant 1 and the one for Plant 2 as shown in Figure 4.21.

production permutation for plant 1

1	1	1	1	1
3	2	7	6	1

production permutation for plant 2

2	2	2
4	5	8

Figure 4.21: Decode Result for the Customer Allocation

These two production permutations, one for each plant, can now be used as the initial solution for the IPDSPPP problem described in Chapter 3. Two IPDSPPP problems will be solved separately using the methods proposed in Chapter 3. After solving these two IPDSPPP problems, the solutions are merged to get the initial solution for the MPIPDSPPP.

The next step is to find a random neighbor of the current customer allocation. As shown in Figure 4.22 and Figure 4.23, two swap operations are used on the encoded representations to randomly identify possible new neighbors. The swap operation has two stages: 1) two plants in the ally permutation trade places and 2) two customers in the production permutation trade places. This is illustrated in Figures 4.22 and .4.23.

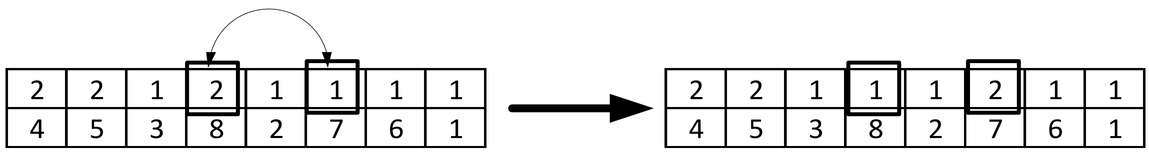

Figure 4.22: First Swap Operation, Just Swap Ally Permutation

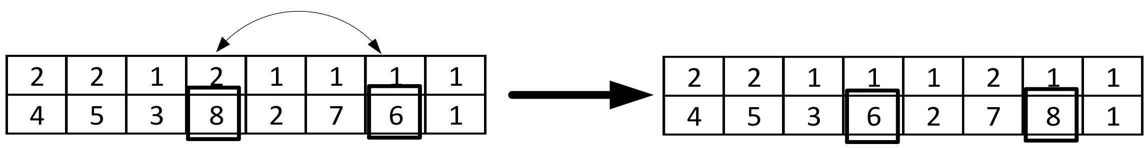

Figure 4.23: Second Swap Operation, Just Swap Production Permutation

It is not hard to imagine that the swap operations can generate infeasible solutions. For example, customer 5 can only be served by plant 2 and customer 2 can only be served by plant 1 so any swap that does not maintain this assignment is infeasible. This is illustrated in Figure 4.24

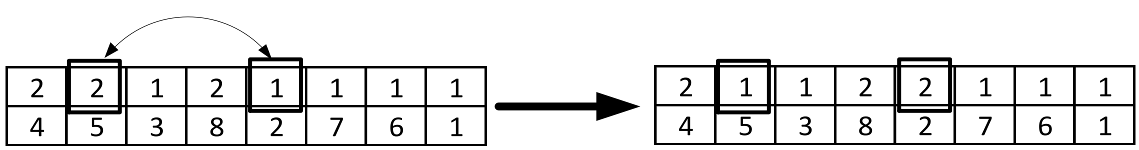

Figure 4.24: Example of Infeasible Representation after Swap Operation

To address this problem, a Value Encoding Method is used to encode the ally permutation. Each ally permutation position will be assigned a random number; the allocation plant will be calculated by a mapping function to map the random number to the customer feasible plant. Suppose the random number is n, SL_j is the servable plant list of customer j, the mapping function is to choose the i^{th} plant in customer j's servable plant list, where i = (n *Mod* $|SL_j|$) + 1. *Mod* is the function to get the reminder. Suppose we have a production permutation item with value 8, its ally permutation value is 3. We know that customer 8 can be served by plant 1 and plant 2. According the mapping function, we will choose the i^{th} plant in the servable plant list, where i = (3 *Mod* 2) + 1 = 2. The customer 8 will be assign to plant 2. Figure 4.25 shows a full Encoded Representation of this example.

ally permutation	42	21	11	3	5	12	4	2
production permutation	4	5	3	8	2	7	6	1
Serviable plant list	{2}	{2}	{1}	{1,2}	{1}	{1,2}	{1,2}	{1}

=

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

Figure 4.25: An Encoding Representation of Example 4.7

As illustrated in Figure 4.25 any random number in ally permutation can be transformed to a servable plant to the any customer with the value encoding method. This method prevents infeasible solutions.

Once the encoding and neighborhood function are determined and the allocation result obtained, the problem is reduced to multiple IPDSPPP problem could be solved by the heuristic provide in Chapter 3.

Numerical Analysis

Earlier in this chapter, several heuristics methods for customer allocation were proposed for separating one MPIPDSPPP into multiple IPDSPPP problems, subsequently solving each of the latter separately. This section analyzes the experimental study conducted using a numerical example to demonstrate the quality of the customer allocation methods. The performances of these methods are compared with one another based on the objective value of the solution generated for the MPIPDSPPP problem.

As discussed earlier, the heuristic allocation method only can assign the customers into plants. After the allocation, the newly generated IPDSPPP problems require the heuristics proposed in Chapter 3 to solve them. In this analysis, two algorithms from Chapter 3 were used to solve the IPDSPPP problem, i.e. the H4 + SA search heuristic was used to solve the IPDSPPP with a large size problem (customers>7), while the H5 + SA search heuristic was used to solve the IPDSPPP with a small size problem (customers≤7). Furthermore, because of the randomness of heuristic as discussed earlier, 50 replications were run for each heuristic algorithm.

The first analysis (B1), which checked the quality of the customer allocation methods, is similar to the A2 analysis in Chapter 3. While the NP-hard complexity of the MPIPDSPPP justifies the use of a heuristic algorithm to solve the problem, providing a measure of the solution's quality is also very important. As with the A2 analysis, the optimality gap was also used here to measure the quality of the heuristic allocation methods.

$$optimality\ gap = \frac{Best\ Heuristic\ Solution - optimal\ solution}{optimal\ solution}$$

The MIP model with current software and hardware was used to obtain the optimal solution, and the 7 customer allocation methods, for the heuristic solution. The best result from the 7 heuristics was named and saved as the Best Heuristic Solution. As discussed earlier, with current software and hardware can only support the optimal solution for the MPIPDSPPP with 7 or fewer customers in the system. In total, 15 test problems were generated for this analysis, the customer number in the system being as the measure for creating 5 groups of test problems (i.e. customer number = 3, 4, 5, 6, 7). Each group included 3 test problems with the same customer number but random locations and demands.

The second analysis (B2), which was the performance comparison between different heuristic allocation methods, was similar to the A3 analysis in Chapter 3. Each heuristic allocation method ran 50 replications for each test problem to determine the minimum, maximum and average cost objective values. In total, 16 test problems involving different customer numbers were tested using the 7 allocation methods, the customer number in the test problem increasing from 10 to 160. The IPDSPPP problem in this analysis was solved using a simulated annealing search and the H4 heuristic algorithm.

The third analysis (B3) explored the impact of input parameters on the heuristic performance. The product lifetime, the plant production rate, the fixed cost for each vehicle, and the variable cost for each vehicle were analyzed. For the MPIPDSPPP with multiple plants, each plant production rate was considered as an individual input parameter. This analysis considered a 2-plant scenario, the test input parameters being the product lifetime, Plant 1 production rate, Plant 2 production rate, the fixed cost for each vehicle and the variable cost for each vehicle.

Experimental Results

1. B1 Analysis

The B1 analysis, similar to the A2 analysis in Chapter 3, was used to investigate the quality of the heuristic customer allocation method. Its best solution by using heuristics, i.e. the Best Heuristic Solution, is the best one for all heuristics across all runs for the test problem; in this research, it was the best solution among 7 heuristics * 50 runs = 350 runs. The optimality gap used to measure the quality of heuristic solution is represented by

$$optimality\ gap = \frac{Best\ Heuristic\ Solution - optimal\ solution}{optimal\ solution}$$

The basic input parameters for the 15 test problems are given in Table 4.9. As described earlier, the optimal solution was generated by the MIP model with AMPL + GRUOBI. The heuristic solution was generated by different allocation methods + simulated annealing search with the H5 heuristic. The results for all runs are shown in Table 4.10.

Table 4.9: Input Parameters for B1 Analysis

Parameters	Values
Customer demand	$q_i \sim U(1, 15)$
Total number of customer	3, 4, 5, 6, 7
Customer location range	(x, y), 0≤x≤300, 0≤y≤200
Plant location	(100, 100), (200, 100)
Plant production rate	r_1=1, r_2=1
Vehicle capacity	C_1=12, C_2=15
Vehicle variable cost per unit distance	R_1=1, R_2 =1.1
Vehicle fixed cost	F_1=500, F_2 =600
Time horizon	400

Figure 4.26, which contains data abstracted from Table 4.10, shows the optimality gaps for the 15 test problems. As can be seen in this figure, optimality gaps for all test problems are 0 which indicate that at least one of the heuristic allocation methods

provides the optimal solution when the customer size is small (customer number ≤ 7). However, Table 4.10 indicates that the difference in solving time between the optimal solution and the heuristic solution is significant: the optimal solution requiring a few hours when the problem size increased to 5 customers, while the heuristic algorithm needed only a few seconds. Furthermore, the time for the heuristic algorithm is the total sum time for running 350 runs (7 heuristics * 50 runs per heuristic).

Table 4.10: All Running Results for B2 Analysis

Problem Number	Customer Number	Optimal Solution	Solving Time (optimal)	Best Heuristic Allocation	Solving Time (heuristic)
1	3	793	72	793	7.482
2	3	982	50	982	9.296
3	3	1176.4	38	1176.4	6.902
4	4	1240.2	51	1240.2	10.268
5	4	1651	94	1651	10.665
6	4	1639.6	234	1639.6	8.970
7	5	1571	72176	1571	13.104
8	5	2096.8	82125	2096.8	13.041
9	5	1730.8	62864	1730.8	9.636
10	6	1744	50413	1744	13.338
11	6	1888.6	126288	1888.6	12.954
12	6	1732.8	93040	1732.8	13.551
13	7	2017.3	78475	2017.3	15.662
14	7	2711.8	131186	2711.8	14.066
15	7	1556	128647	1556	13.631

Note: the unit for solving time is seconds.

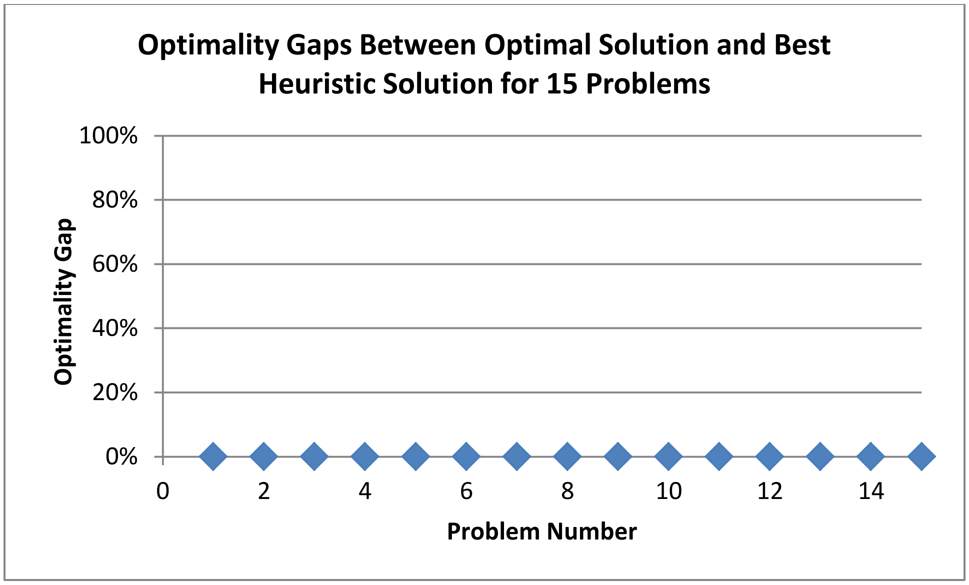

Figure 4.26: Optimality Gaps Between Optimal Solution and Best Heuristic Solution for 15 Test Problems in B1 Analysis

Figure 4.27 shows the number of times each allocation method provided the Best Heuristic Solution, indicating that the SA allocation method provided it every time while the fixed allocation methods (M1, M2, M3 and M4) provided it fewer than twice, probably because the structures of these heuristics are different. The fixed allocation method can only provide one customer allocation for each test problem. When this allocation does not fit the one that can lead to the optimal solution, fixed allocation methods never reach the optimal solution for that test problem. The remaining 2 random allocation methods (M5 and M6) do not include a learning strategy, meaning they have only 50 chances (i.e. 50 runs) to reach the best allocation solution for each problem. The SA allocation method, which uses a learning strategy to change the customer allocation within each run, can search more customer allocations to find the better solutions. Table

4.11 provides the detailed running time for each heuristic, while the pie chart in Figure 4.28 illustrates the proportion of time needed for each algorithm to reach the Best Heuristic Solution, indicating that the SA method runs much longer than the other heuristics (≥80% of time).

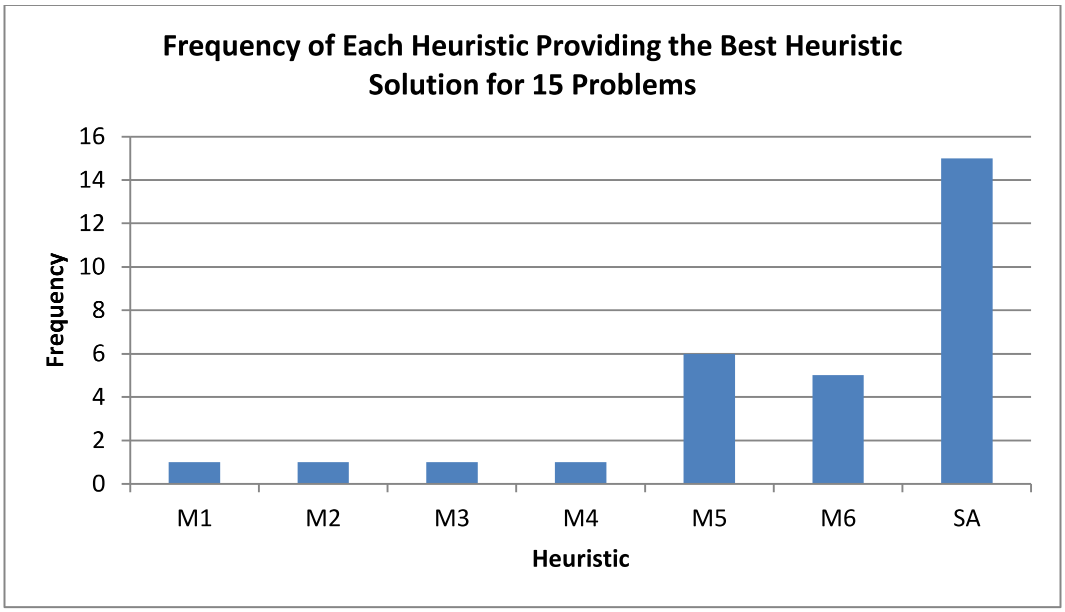

Figure 4.27: Frequency of Each Heuristic Providing the Best Heuristic Solution for 15 Problems

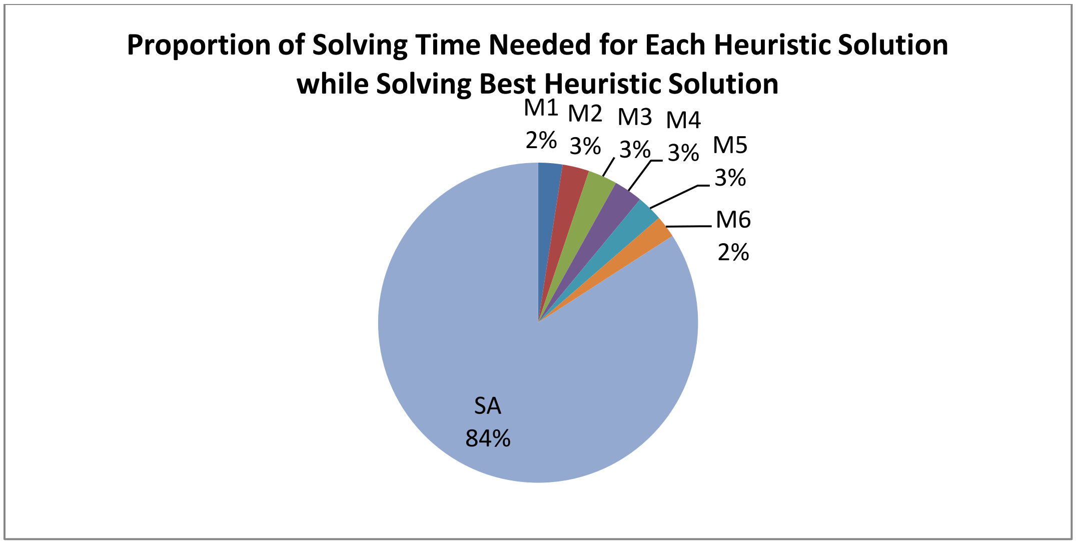

Figure 4.28: Proportion of Solving Time Needed for Each Heuristic Solution While Solving Best Heuristic Solution

Table 4.11: Running Times for All Customer Allocation Methods for the 16 Problems

Problem Number	M1	M2	M3	M4	M5	M6	SA	Best Heuristic Solution
1	0.090	0.345	0.240	0.285	0.240	0.255	6.027	7.482
2	0.270	0.330	0.315	0.360	0.345	0.240	7.436	9.296
3	0.255	0.360	0.225	0.315	0.270	0.240	5.237	6.902
4	0.270	0.285	0.300	0.270	0.225	0.360	8.558	10.268
5	0.390	0.285	0.360	0.435	0.225	0.210	8.760	10.665
6	0.270	0.330	0.360	0.345	0.255	0.285	7.125	8.970
7	0.390	0.390	0.300	0.240	0.330	0.240	11.214	13.104
8	0.375	0.360	0.405	0.285	0.255	0.225	11.136	13.041
9	0.315	0.255	0.270	0.285	0.270	0.315	7.926	9.636
10	0.240	0.300	0.450	0.345	0.390	0.390	11.223	13.338
11	0.225	0.285	0.345	0.300	0.360	0.300	11.139	12.954
12	0.285	0.195	0.225	0.375	0.345	0.360	11.766	13.551
13	0.510	0.285	0.360	0.360	0.360	0.210	13.577	15.662
14	0.150	0.270	0.435	0.345	0.195	0.060	12.611	14.066
15	0.210	0.375	0.480	0.480	0.480	0.180	11.426	13.631
Sum	4.245	4.650	5.070	5.025	4.545	3.870	145.158	172.563

Note: the time is measured in seconds. Each record shows the sum time of 50 runs.

2. B2 Analysis.

Analysis B2, which is similar to the A3 analysis in Chapter 3, compared the 7 customer allocation methods against one another based on the objective value of the solution of the MPIPDSPPP problem. In this analysis, a total of 16 problems were tested, each having different customer sizes, demands and locations. All test problems were large size problems (customer size > 7) with the number of customers increasing from 10 to 160. The rest of the input parameters are shown in Table 4.12. Similar to the B1 analysis, each heuristic ran 50 times for each problem to obtain the results. However, unlike for the B1 analysis, not only the minimum cost but also the average cost and the maximum cost for the 50 runs were recorded for comparison.

Table 4.12: Input Parameters for B2 Analysis

Parameters	Values
Customer demand	$q_i \sim U(1, 15)$
Customer location range	(x, y), $0 \le x \le 300$, $0 \le y \le 200$
Plant location	(100, 100), (200, 100)
Plant production rate	r_1=1, r_2=1
Vehicle capacity	C_1=12, C_2=15
Vehicle variable cost per unit distance	R_1=1, R_2 =1.1
Vehicle fixed cost	F_1=500, F_2 =600
Time horizon	1000

All running results are reported in Tables 4.13, 4.14 and 4.15, with Figures 4.29, 4.30, 4.31 presenting their abstracted data, the results shown in the figures are ordered by problem number, which was also the order of the customer numbers in the system. When the problem number increased, the number of customers in the system increased with a constant rate. As these three tables show, all three costs (average, minimum and maximum) are increased as the customer numbers increased. This observation is obvious. A logical conclusion for this observation is that more customers in the system require higher transportation costs. Form these figures, we could see that the methods M5 and M6 usually provider higher cost in all three criteria (average, minimum and maximum). The comparison for rest methods (M1, M2, M3, M4, SA) need more sophisticated statistical analysis.

Table 4.13: Running Result for the B2 Analysis with the Average Objective Value for 16 Problems

Problem Number	Customer Number	Average Objective Value						
		M1	M2	M3	M4	M5	M6	SA
1	10	3097.8	3587.6	2933.8	3107.2	4041.2	3877.8	2467.6
2	20	5745.9	5900.5	5252.8	5238.6	6927.8	7835.6	5192.7
3	30	7156.6	7504.6	6961.3	7221.9	9049.9	9673.0	6810.7
4	40	8086.0	8487.5	8397.3	7966.2	12185.5	11760.2	8361.4
5	50	11166.1	11643.8	11284.8	12044.5	16805.4	16460.6	10995.4
6	60	12559.5	13113.6	13282.9	13138.1	19835.1	20884.0	11997.3
7	70	14802.5	14958.3	15011.7	15318.7	22281.8	21277.6	14843.7
8	80	16047.1	16072.8	16128.3	15877.5	22887.4	24423.7	15631.0
9	90	19713.1	19660.0	19964.2	19656.3	29106.9	26152.4	19231.4
10	100	18997.4	19365.3	19114.1	19200.4	30911.3	29363.6	19235.2
11	110	21727.8	21596.2	21765.2	22280.8	36417.9	35122.1	21853.3
12	120	25425.6	27249.1	27222.1	26309.0	37403.9	34423.4	25433.7
13	130	25174.8	27572.6	25343.5	24261.6	36605.8	35417.3	24839.0
14	140	27442.5	28343.8	27842.7	27626.1	43836.9	43143.5	27065.6
15	150	29495.9	29766.1	28692.6	29538.8	43283.5	41817.0	28673.2
16	160	33569.3	38005.2	34155.6	34117.8	50026.1	47664.5	33044.8

Table 4.14: Running Result for the B2 Analysis with the Maximum Objective Value for 16 Problems

Problem Number	Customer Number	Maximum Objective Value						
		M1	M2	M3	M4	M5	M6	SA
1	10	3109.2	3592.8	3197.0	3107.9	4313.2	3914.0	2706.4
2	20	5874.8	5922.8	5269.8	5297.8	7921.5	8245.6	5297.5
3	30	7194.2	7641.6	7092.1	7397.6	10369.0	9766.8	7043.0
4	40	8402.0	8531.6	8455.8	8246.0	12639.5	12481.5	8892.8
5	50	11618.6	12153.6	11512.5	12149.0	16956.4	17000.9	11370.7
6	60	13360.9	13342.4	13703.2	13787.3	19979.6	21678.0	13324.7
7	70	14959.0	15397.0	15210.5	15408.3	23300.6	21842.9	15113.9
8	80	16881.2	16559.9	17003.2	15934.7	23933.3	24972.2	16074.2
9	90	20295.2	19774.3	20450.7	20345.0	30262.1	27185.9	19570.3
10	100	19425.6	19642.4	19501.1	19712.6	31798.0	30095.9	20315.0
11	110	22178.1	22274.4	22178.1	22641.5	37108.2	35891.1	22404.0
12	120	25745.6	27489.8	27356.7	27087.8	38627.4	39634.9	25883.4
13	130	25727.8	28225.2	26182.4	24719.0	38480.5	38074.3	25207.5
14	140	27719.9	28735.6	28148.5	29312.9	44963.9	44304.7	27454.6
15	150	30394.5	30300.4	29469.8	30105.3	44030.1	42701.4	29534.9
16	160	34440.5	39183.1	34853.7	35437.4	51069.6	48515.6	33862.0

Table 4.15: Running Result for the B2 Analysis with the Minimum Objective Value for 16 Problems

Problem Number	Customer Number	Minimum Objective Value						
		M1	M2	M3	M4	M5	M6	SA
1	10	3096.5	3577.0	3197.0	3107.9	4313.2	3914.0	2706.4
2	20	5212.8	5272.7	5269.8	5297.8	7921.5	8245.6	5297.5
3	30	6449.8	7067.6	7092.1	7397.6	10369.0	9766.8	7043.0
4	40	7733.5	8408.0	8455.8	8246.0	12639.5	12481.5	8892.8
5	50	10663.6	10777.6	11512.5	12149.0	16956.4	17000.9	11370.7
6	60	12355.2	12565.6	13703.2	13787.3	19979.6	21678.0	13324.7
7	70	14307.4	14553.9	15210.5	15408.3	23300.6	21842.9	15113.9
8	80	15640.2	15758.9	17003.2	15934.7	23933.3	24972.2	16074.2
9	90	18978.6	18964.0	20450.7	20345.0	30262.1	27185.9	19570.3
10	100	18485.6	18819.4	19501.1	19712.6	31798.0	30095.9	20315.0
11	110	20730.4	20915.3	22178.1	22641.5	37108.2	35891.1	22404.0
12	120	24905.0	26023.6	27356.7	27087.8	38627.4	39634.9	25883.4
13	130	24061.2	26847.4	26182.4	24719.0	38480.5	38074.3	25207.5
14	140	27029.0	27796.3	28148.5	29312.9	44963.9	44304.7	27454.6
15	150	28725.9	29139.3	29469.8	30105.3	44030.1	42701.4	29534.9
16	160	32629.6	34917.8	34853.7	35437.4	51069.6	48515.6	33862.0

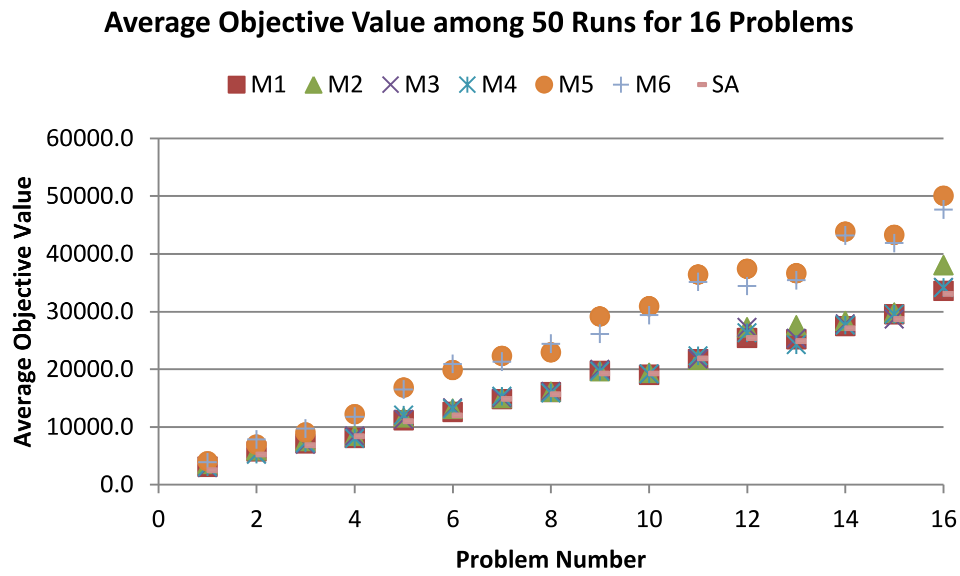

Figure 4.29: Comparison of the Average Objective Values of the Heuristics for 16 Problems in the B2 Analysis

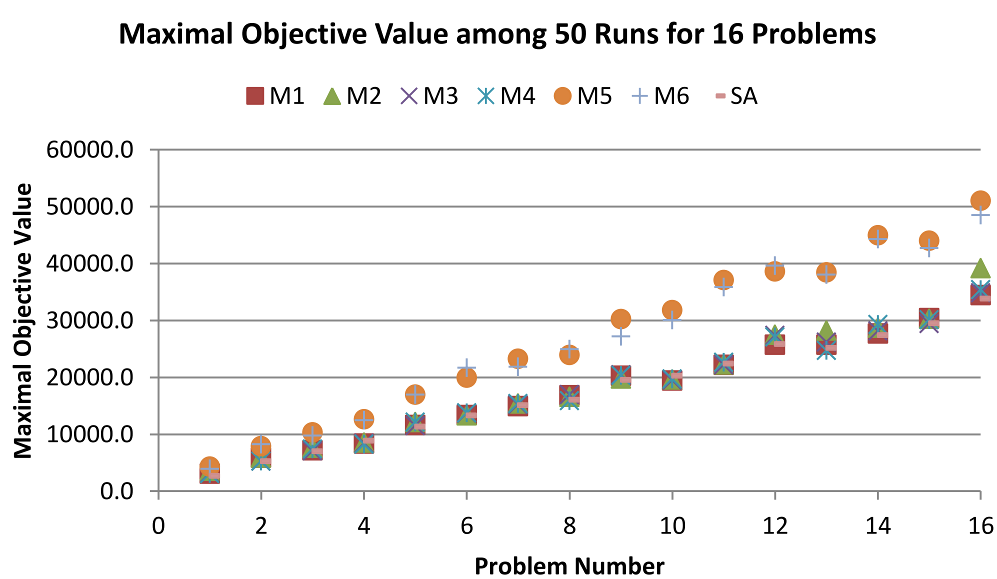

Figure 4.30: Comparison of the Maximum Objective Values of Heuristics for 16 Problems in the B2 Analysis

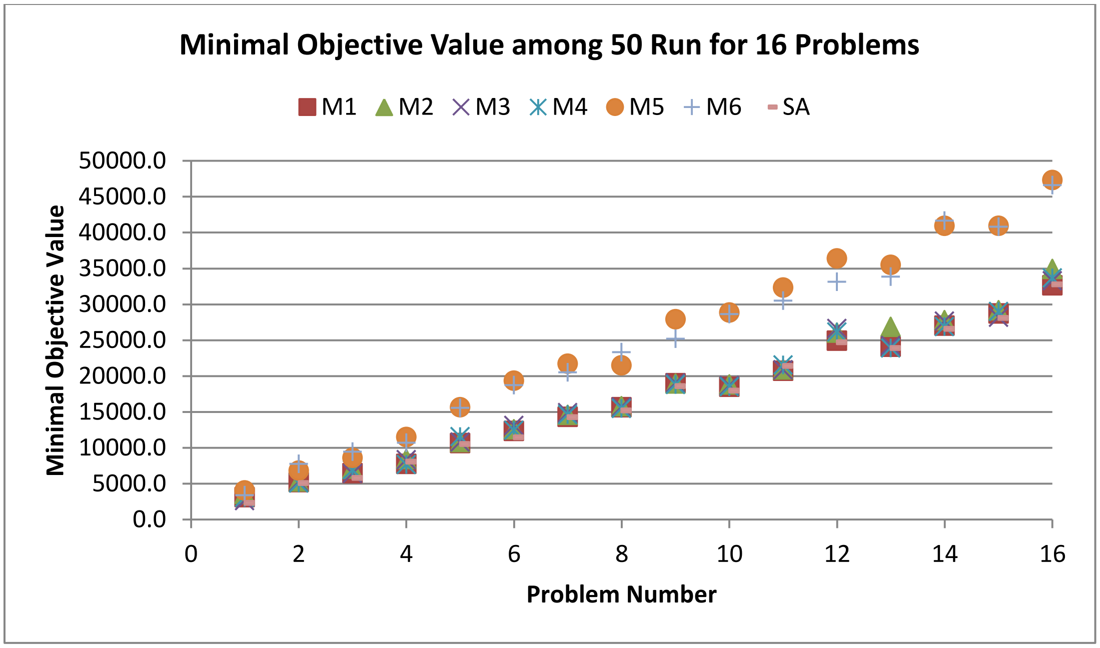

Figure 4.31: Comparison of the Minimum Objective Values of Heuristics for 16 Problems in the B2 Analysis

To more clearly see their performance, Figures 4.32, 4.33 and 4.34 display the number of times each of the 7 heuristics arrived at the best solutions for minimum cost, maximum cost and average cost, respectively. The SA heuristic performed the best for these three criteria (minimum cost, average cost, maximum cost). On the other hand, the performance of the remaining 2 random allocation heuristics (M5 and M6) was not good as none achieved the best solution for any of the criteria for any test problem. The following statistical analyses support the conclusion that the SA heuristic performed better than the rest of the heuristics.

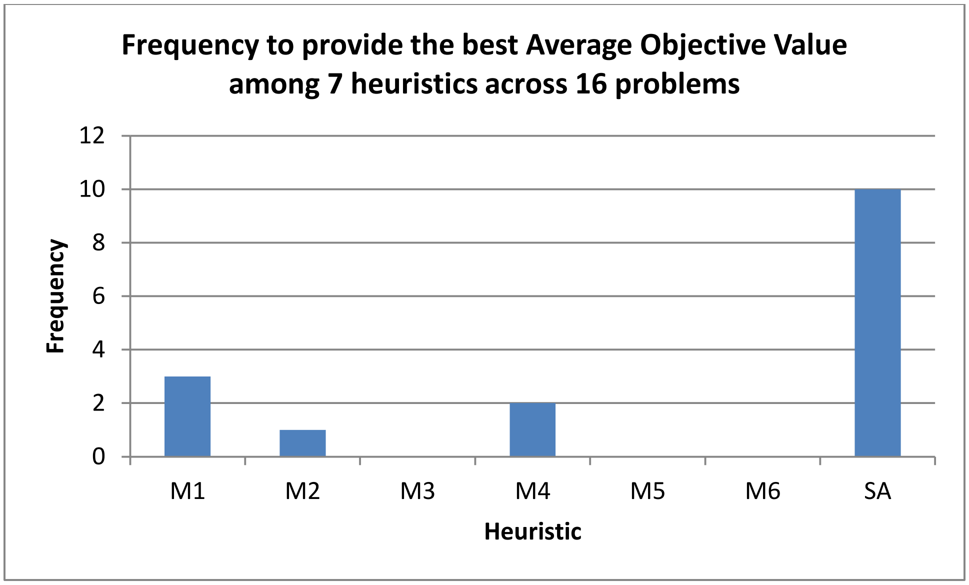

Figure 4.32: Frequency for Providing the Best Average Objective Value among 7 Heuristics across 16 Problems

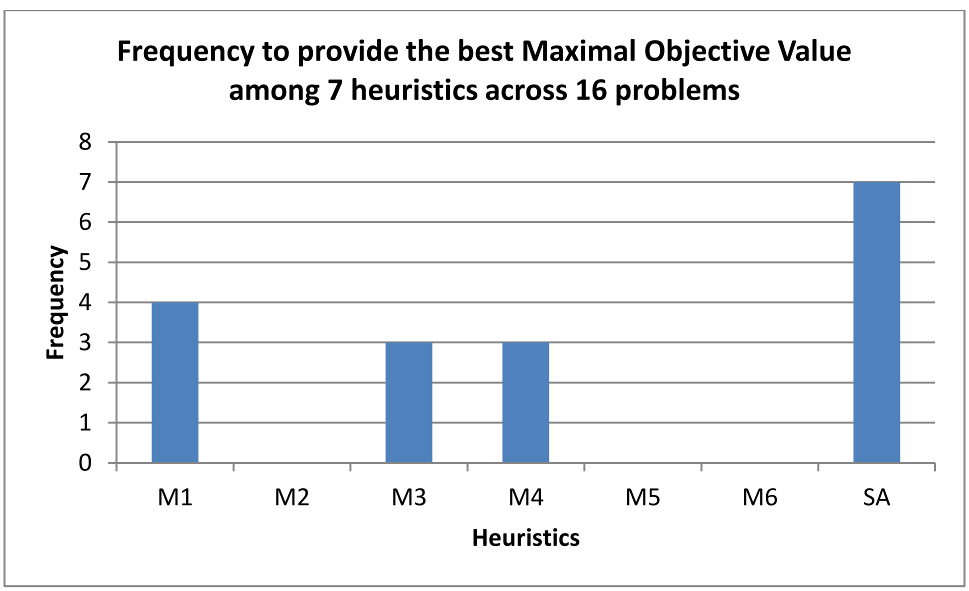

Figure 4.33: Frequency for Providing the Best Maximum Objective Value among 7 Heuristics across 16 Problems

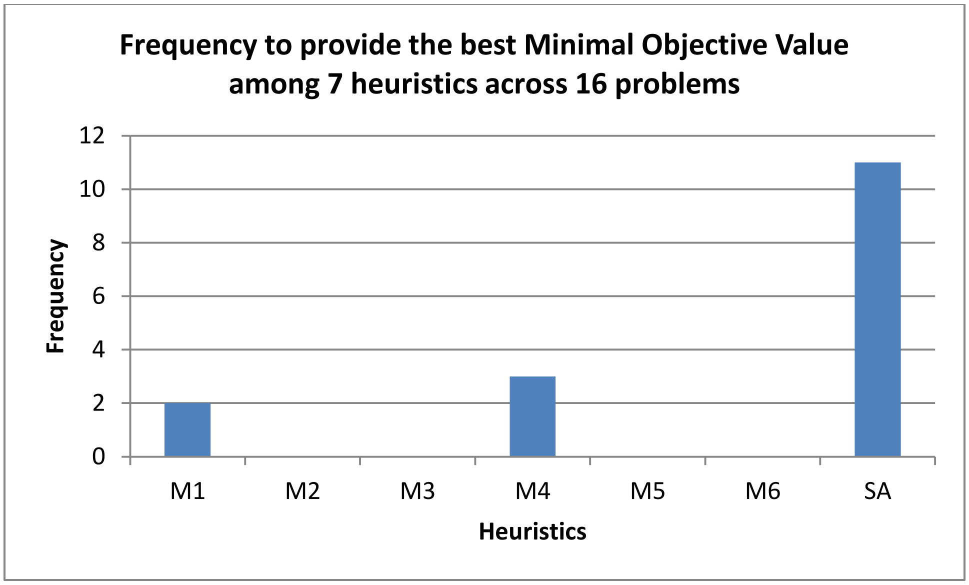

Figure 4.34: Frequency for Providing the Best Minimum Objective Value among 7 Heuristics across 16 Problems

The Anderson-Darling test was again used to test the normality of the raw data shown in Tables 4.13, 4.14 and 4.15, the results seen in the Appendix D plots of Tables D.1, D.2, and D.3 indicating that these data did not exhibit a normality distribution. As a result, nonparametric statistical tests were required. Furthermore, Figures 4.29, 4.30 and 4.31 suggest the customer number in the system exhibits a linear relationship with the total system transportation cost.

Similar to the A3 analysis, the Freidman test was used to determine whether the heuristic method affected the quality of the solution, the 7 heuristics being the first factor in the analysis and the customer numbers the second. Three Freidman Tests were conducted, the hypotheses being shown in Table 4.16.

Table 4.16: Hypotheses for All Tests in B2 Analysis

B2 Test1	H0: medians of the minimum cost are equal for all heuristics H1: at least one is median is different
B2 Test2	H0: medians of the maximum objective values of different heuristics are equal H1: not all medians of the maximum objective values of different heuristics are equal
B2 Test3	H0: medians of the average objective values of different heuristics are equal H1: not all medians of the average objective values of different heuristics are equal

Appendix Figures B.4, B.5 and B.6 present the statistical results of the Freidman tests with R, all exhibiting a small *p*-value (<0.001) meaning there is sufficient evidence to reject the null hypothesis and conclude that at least one median value from each test is different from the others. The post-hoc test result indicates that the SA heuristic is significant different from the M2, M3, M4, M5 and M6 heuristics for all three tests. In addition, the SA is significantly different from M1 for the minimum objective value test, but not significantly different from M1 for the other two tests (average cost test and maximum cost test). The sum ranks of all heuristics for the three Freidman Test are abstracted in Table 4.17, with Figure 4.33 displaying these data indicating that the SA method does have the lowest sum ranks for all three criteria (minimum, average, maximum cost).

Table 4.17: Sum Ranks of All Heuristic for the Three Freidman Test

	M1	M2	M3	M4	M5	M6	SA
Avg	38	68	56	51	108	100	27
max	41.5	64	47.5	54	107	101	33
min	41	70	59	47	108	99	24

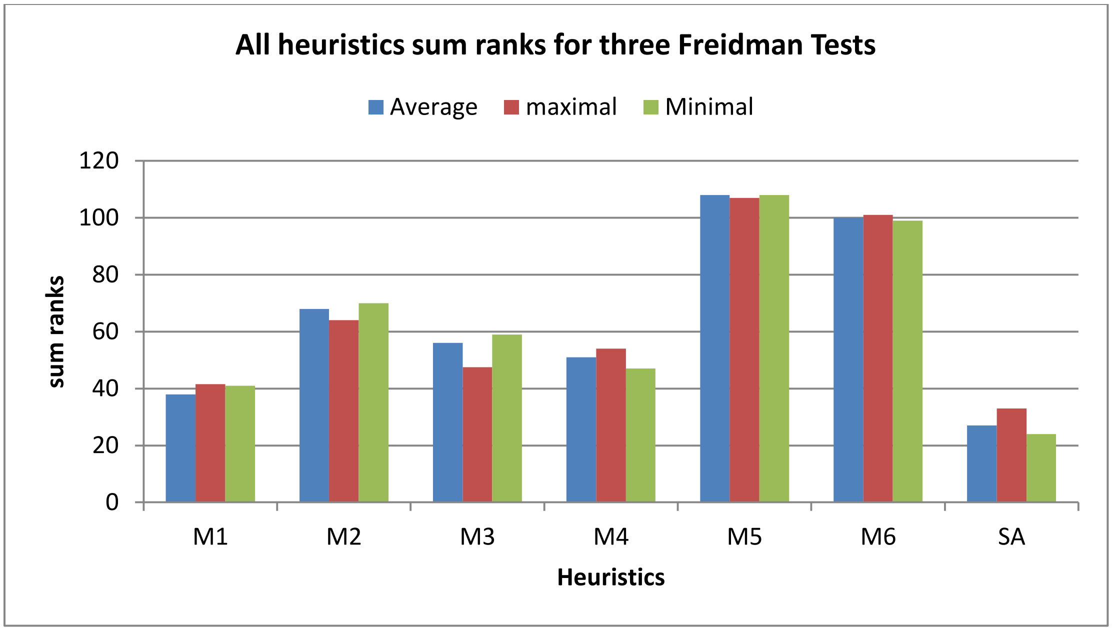

Figure 4.35: Sum Ranks of All Heuristic for the Three Freidman Test

For the Freidman test, the best solution is ranked as value 1, the second best as value 2, and so on, meaning that because of its lower sum ranks and its significant differences from the other heuristics, the SA algorithm performed the best among the 7 heuristics. In addition, the analysis results indicated M1 also performed well, exhibiting no significant differences from the SA method for the test criteria of maximum cost and average cost. These results suggest that the worst case and average performance of M1 method is similar to that of the SA method, which was found to be better than other methods. Since the M1 method assigned the customer to the nearest plant, its good performance indicates that the direct distance to the plant has a large impact on the decision of customer allocation. In addition, the two random methods (M5 and M6) exhibited the worst performance, indicating that chance and balancing the plant's production capacity are not as important as the direct distance when deciding customer allocation.

3. B3 analysis

In the previous analysis, the test problems differed only in the number of customers in the system, customer location and demand, the other input parameters remaining the same. The B3 analysis investigated the sensitivity of the solution to changes in product lifetime, vehicle fixed cost, vehicle variable cost, and plant production rate. Since the current problem involves multiple plants, each plant was treated separately, meaning the production rates for Plants 1 and 2 were considered as 2 separate factors. This analysis uses a 100 customers' problem as the base problem. The base input parameters for this problem are shown in Table 4.18. The problem will be solved using the SA method.

Table 4.18: Base Input Parameters for B3 Analysis

Parameters	Values
Customer demand	$q_i \sim U(1, 15)$
Customer location range	(x, y), $0 \le x \le 300$, $0 \le y \le 200$
Plant location	(100, 100), (200, 100)
Plant production rate	$r_1=1$, $r_2=1$
Vehicle capacity	$C_1=12$, $C_2=15$
Vehicle variable cost per unit distance	$R_1=1$, $R_2=1.1$
Vehicle fixed cost	$F_1=500$, $F_2=600$
Time horizon	800

This analysis is similar to A4 in Chapter 3 for which each test input parameter generated a group of sub-problems by varying ±90% from the base value while the other input parameters remained at the base value. The sensitivity analysis running results are shown in Appendix E, Table E.1.

Figure 4.36 plots the data from Table D.1 showing the minimum objective function value for the system transportation cost based on the 5 input parameters. As shown in this figure, the change in both plants' production rates and product lifetimes exhibit no discernible relationship with transportation cost, and the fixed cost and variable cost appear to exhibit a linear relationship with the total system transportation cost. To confirm this finding, analysis A4 employed a correlation analysis to test whether there was a correlation between input parameters (variable vehicle cost and the fixed vehicle cost) and the total system transportation cost. The raw data were once again checked for normality using the Anderson-Darling test; resulting plots can be found in Appendix E.2 and E.3. The results indicate that the data for variable cost and fixed vehicle cost followed a normal distribution. As a result, two 2 Pearson Correlation Tests were conducted to examine the correlation between the input parameters and the objective values. The hypotheses are shown in Table 4.19.

Table 4.19: Hypotheses of All Tests in B3 Analysis

B3 Test1	H0: the correlation between variable vehicle cost and system transportation cost is 0 H1: the correlation between variable vehicle cost and system transportation cost is not 0
B3 Test2	H0: the correlation between fixed vehicle cost and system transportation cost is 0 H1: the correlation between fixed vehicle cost and system transportation cost is not 0

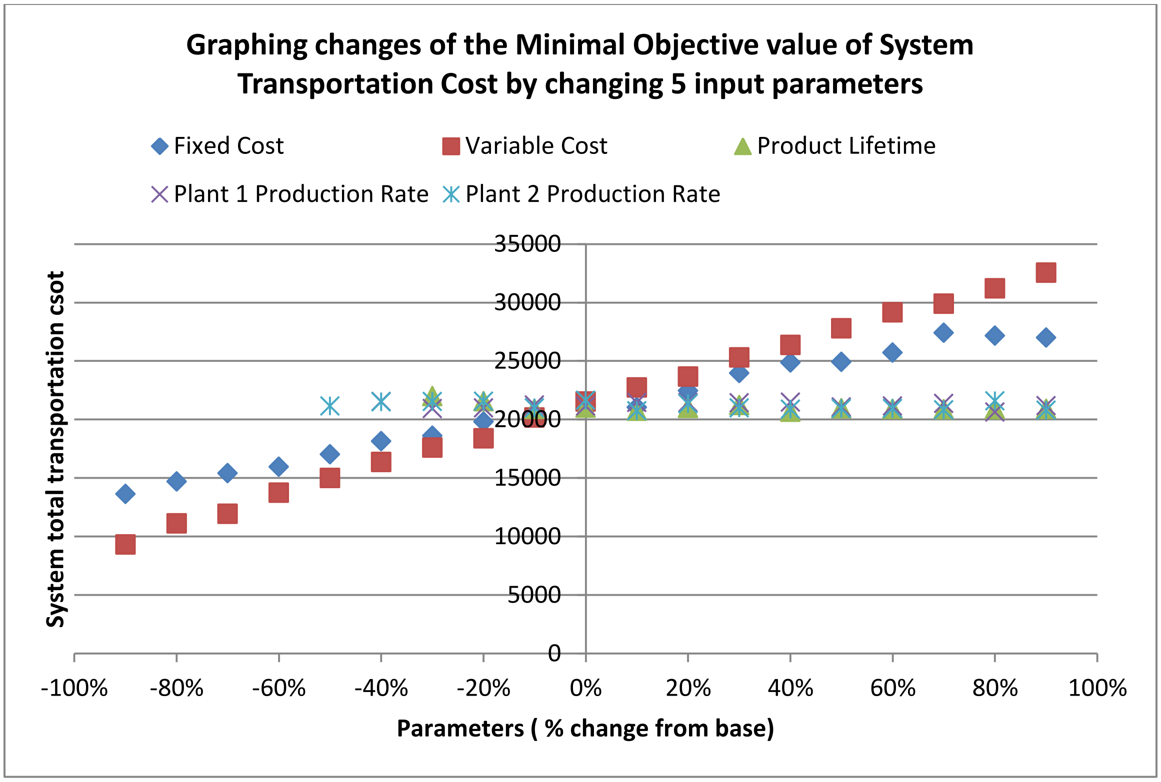

Figure 4.36: Changes in the Minimum Objective Value of System Transportation Cost in Relation to the Changes in the 5 Input Parameters

Both correlation test results, which are shown in Appendices E.4 and E.5, indicate sufficient evidence to reject the null hypotheses, meaning the variables vehicle cost and fixed vehicle cost exhibit a correlation with the system total transportation cost. Furthermore, this correlation is a strong positive linear relationship (correlation >=0.99). The results of B3 statistical analysis are consistent with those from the A4 analysis: the variable vehicle cost and fixed vehicle cost exhibit a positive linear relationship with the system total transportation cost, as expected. The remaining input parameters, the production lifetime and the plant production rate, control the feasibility of the solution rather than the total system cost. As shown in Table E.1 in Appendix E, no feasible solution was found for the problem when the product lifetime was below 70% of the base value, the production rate of Plant 1 less than 60% of base value, or the production rate of Plant 2 rate less than 50% of base value.

Conclusion and Further research

The multiple plant integrated production and distribution scheduling problem was analyzed in this chapter. The key function, customer allocation method was discussed in detail, and the formula for calculating the total number of customer allocations was provided. In addition, both fixed and random customer allocation methods were developed. Based on the results from the statistical analysis, the Simulated Annealing method exhibited the best performance of the methods investigated, and the input parameter analysis indicated that the system total transportation cost exhibits a linear relationship with the vehicle fixed cost and variable cost, while the product lifetime and plant production rate control the feasibility of the solution.

Further research could involve investigating practical extensions of the IPDSPPP problem. One such extension could apply MPIPDSPPP variables such as delivery involving a time window, stochastic customer demand, and split delivery in the transportation component.

In addition, there are several unique extensions of the MPIPDSPPP problem, one being relaxing the constraint of not being able to share vehicles among plants in the multi-plant scenario. It is usually more profitable to use a smaller fleet with longer transportation times than a larger fleet with less transportation time for the distribution function when the vehicle fixed cost is high (e.g. labor cost and vehicle rent cost). For example, if the planning horizon is one week and Plant 1 requires 2 vehicles for 3 days to complete distribution and Plant 2 requires 2 vehicles for 2 days, sharing the vehicles would involve using 2 vehicles to complete the distribution for both plants in 4 days. Using two fewer vehicles would result in fixed variable cost saving. Though the variable cost (travelling from Plant 1 to Plant 2) might increase, the saving in the fixed variable cost would still result in a lower total transportation cost.

Another realistic extension for the multi-plant case is to relax the assumption that all vehicles must replenish the product at the plant where it begins delivery. Doing so would allow the vehicle to save variable cost by replenishing the product at the closer plant rather than returning to the original one. Furthermore, the system performance might improve. For instance, a vehicle from Plant 1 has completed a trip delivery and is near Plant 2 when it is ready to be replenished; at the same time, all the vehicles from Plant 2 are not yet ready to be replenished. By replenishing at Plant 2, this vehicle can help finish one delivery trip for that plant, or it could replenish the product but continue distributing for Plant 1. While relaxing this assumption may improve performance, it increases the complexity of the problem significantly.

CHAPTER FIVE

CONCLUSION

The integrated production and distribution scheduling problem such as the one investigated here is an NP-hard problem. Specifically, this dissertation discussed a practical extension of the IPDSP considering the perishable product using both a single plant and a multiple plant situation. The mathematical models for both situations were developed and solved optimally using the mixed integer programming model for small size problems (i.e. less or equal to 7 customers both in single plant situation and multiple plants situation); however, this programming model is not effective when the problem size increases; it does not result in a feasible solution when the problem size is more than 20 customers in single plant situation and more than 15 customers in a multiple plant situation. Since an NP-hard problem was analyzed here, heuristic algorithms were implemented for both the single plant and multiple plant situations. The benefits of using heuristics include 1) the ability to solve a large problem which cannot be handled by the mixed integer programming model, and 2) the short computation time. A single plant problem with 100 customers was solved using the Multi-stop Delivery with Random Shortest Path Fit algorithm (H4) within 2 minutes.

For the single plant situation, five heuristics were created as discussed in Chapter 3 to solve the IPDSPPP problem. Furthermore, an improved LB was conducted for the problem, one that considered the internal traveling distance between customers to improve travel distance, and accumulating the waiting time for all available vehicles at the beginning of the time horizon to improve used the vehicle number. The statistical analysis suggested that the new LB performance was better than the former one. In addition, the performance of the heuristics was analyzed. The heuristic algorithms resulted in the optimal or near optimal solution (i.e. gap between the solution and the optimal solution was less than 5%) for the small size problem (i.e. less than or equal to 7 customers in the system). The statistical analysis indicated that H3 and H4 performance was better than the heuristics H1, H2 and H5. In addition, the sensitivity of the objective value to the input parameters was also analyzed, the results showing that the fixed vehicle cost and variable vehicle cost exhibited a linear relationship with the system total transportation cost, while the product lifetime and plant production rate affected only the feasibility of the solution and had little effect on the total transportation cost.

The six heuristic algorithms created to solve the multiple plants IPDSPPP problems were discussed in Chapter 4. Furthermore, the customer allocation problem, the sub-problem of the MPIPDSPPP, was found to be an NP problem. The formula for calculating the total number of customer allocations is also provided. The performance of the heuristic algorithms for the multiple plants scenario was found to good, the statistical results indicating they provided the optimal solution or one with less than a 5% difference from the optimal solution. In addition, the performance among the various heuristic algorithms was compared, the results showing that the SA-based search heuristic (A6) performed better than heuristics A1, A2, A3, A4 and A5.

Future Research

This study, however, is only the first in this research. Below are suggestions for future studies in this area.

First the research could be extended by adding time windows for the distribution. This research assumed delivery could occur anytime within the time horizon. This is true for some customers, like the food industry where the vehicle can deliver the food to the restaurants anytime during the day. However, in other situations the customers have a distinct time window for accepting delivery, one depending on various other business requirements such as the space available in the parking lot or the schedule of the production line. The IPDSPPP problem could be extended to provide a distinct time window for each customer within the time horizon, thereby increasing the complexity of the problem significantly.

Second, the research could be extended by allowing split delivery. In the research reported here, all customer demand was assumed to be satisfied by one vehicle within one trip. However, it would be more practical to allow for split delivery, the demand of a customer being satisfied by 2 or more vehicles. By releasing this constraint, the transportation system could be reduced by using an increased number of full-truck load deliveries, resulting in better utilization of the vehicles and fewer of them.

Third, the research could be extended by allowing vehicles to replenish at any plant, not being assigned to only one. This extension is specific to the multiple plant situation. This research assumed the fleet used by Plant A could not be used to satisfy the distribution for Plant B. In addition, the vehicle could replenish only at the plant it was assigned to. However, since the multiple plants were all owned by the same decision maker, the fleets could work together. Releasing this constraint would allow a vehicle to make multiple trips because it could replenish at various plants as well as return to any one of them at the end of the planning horizon. This will provide more flexibility for the trip decisions, while the same time introducing more complexity into the distribution schedule for each vehicle and the relative production schedule for each plant.

Forth, the heuristic vehicle routing and scheduling decisions developed in this dissertation were based on the route first cluster second heuristic concept. Other heuristics such as cluster first route second (Laporte et al 2000), constructive (Clarke and Wright 1964) and improvement methods (Potvin and Rousseau 1995) could also be used to make decisions for the vehicle routing and scheduling. Furthermore, other meta-heuristic search methods could be applied in looking for the global optimum, including Genetic Algorithms (GA), Tabu Search (TS), and Ant Colony Optimization (ACO), among others.

Furthermore, there are many assumptions can be released in the further work, such as include costs for setup between production batches, considering the inventory cost into the system total cost, considering multiple planning periods and infinite planning periods.

Include inventory costs

Have multiple planning periods

Possibly finding approximate solutions using evolutionary algorithms of the MIP's directly for larger problems

APPENDICES

Appendix A
Statistical Results: Running results for A1 analysis

Table A.1: running result for A1 analysis

Run	Cust. #	Plan Horizon	Old Lower Bound			New Lower Bound			Difference (new - old)		
			LB	Vehicle #	Travel Dist.	LB	Vehicle #	Travel Dist.	LB	Vehicle #	Travel Dist.
1	4	300	685	1	185	686	1	186	1	0	1
2	4	300	624	1	124	654	1	154	30	0	30
3	4	300	734	1	234	777	1	277	43	0	43
4	4	300	608	1	108	676	1	176	68	0	68
5	4	300	642	1	142	675	1	175	33	0	33
6	5	300	698	1	198	750	1	250	52	0	52
7	5	300	782	1	282	1301	2	301	519	1	19
8	5	300	718	1	218	711	1	211	-7	0	-7
9	5	300	1400	2	400	1401	2	401	1	0	1
10	5	300	660	1	160	750	1	250	90	0	90
11	6	300	744	1	244	720	1	220	-24	0	-24
12	6	300	714	1	214	1305	2	305	591	1	91
13	6	300	756	1	256	1299	2	299	543	1	43
14	6	300	1302	2	302	1296	2	296	-6	0	-6
15	6	300	734	1	234	791	1	291	57	0	57
16	7	300	1326	2	326	1337	2	337	11	0	11
17	7	300	668	1	168	696	1	196	28	0	28
18	7	300	746	1	246	1315	2	315	569	1	69
19	7	300	1332	2	332	1365	2	365	33	0	33
20	7	300	772	1	272	1335	2	335	563	1	63
21	8	300	1408	2	408	1433	2	433	25	0	25
22	8	300	750	1	250	1354	2	354	604	1	104
23	8	300	1344	2	344	1376	2	376	32	0	32
24	8	300	1316	2	316	1357	2	357	41	0	41
25	8	300	1356	2	356	1422	2	422	66	0	66
26	9	300	1384	2	384	1444	2	444	60	0	60
27	9	300	1300	2	300	1328	2	328	28	0	28
28	9	300	1360	2	360	1421	2	421	61	0	61
29	9	300	1404	2	404	1466	2	466	62	0	62
30	9	300	1336	2	336	1373	2	373	37	0	37
31	10	300	1320	2	320	1382	2	382	62	0	62
32	10	300	1448	2	448	1525	2	525	77	0	77
33	10	300	720	1	220	771	1	271	51	0	51
34	10	300	1306	2	306	1402	2	402	96	0	96
35	10	300	1404	2	404	1463	2	463	59	0	59
36	20	400	1620	2	620	1726	2	726	106	0	106
37	20	400	2444	3	944	2478	3	978	34	0	34
38	20	400	2450	3	950	2528	3	1028	78	0	78
39	20	400	1690	2	690	2312	3	812	622	1	122
40	20	400	1670	2	670	1759	2	759	89	0	89
41	30	500	2492	3	992	2586	3	1086	94	0	94
42	30	500	1930	2	930	2522	3	1022	592	1	92

43	30	500	1848	2	848	1972	2	972	124	0	124
44	30	500	1868	2	868	2517	3	1017	649	1	149
45	30	500	1918	2	918	2536	3	1036	618	1	118
46	40	600	2048	2	1048	2194	2	1194	146	0	146
47	40	600	2970	3	1470	3093	3	1593	123	0	123
48	40	600	2172	2	1172	2778	3	1278	606	1	106
49	40	600	2070	2	1070	2711	3	1211	641	1	141
50	40	600	2894	3	1394	3026	3	1526	132	0	132
51	50	700	3092	3	1592	3244	3	1744	152	0	152
52	50	700	3142	3	1642	3260	3	1760	118	0	118
53	50	700	3072	3	1572	3241	3	1741	169	0	169
54	50	700	3216	3	1716	3336	3	1836	120	0	120
55	50	700	2290	2	1290	2972	3	1472	682	1	182
56	60	800	3856	3	2356	4482	4	2482	626	1	126
57	60	800	3634	3	2134	3725	3	2225	91	0	91
58	60	800	3842	3	2342	4455	4	2455	613	1	113
59	60	800	3612	3	2112	3739	3	2239	127	0	127
60	60	800	3360	3	1860	3557	3	2057	197	0	197
61	70	900	2780	2	1780	3478	3	1978	698	1	198
62	70	900	3756	3	2256	3946	3	2446	190	0	190
63	70	900	3414	3	1914	3566	3	2066	152	0	152
64	70	900	3398	3	1898	3584	3	2084	186	0	186
65	70	900	3818	3	2318	3946	3	2446	128	0	128
66	80	1000	4296	3	2796	5010	4	3010	714	1	214
67	80	1000	3962	3	2462	4115	3	2615	153	0	153
68	80	1000	4212	3	2712	4346	3	2846	134	0	134
69	80	1000	4096	3	2596	4220	3	2720	124	0	124
70	80	1000	3898	3	2398	4046	3	2546	148	0	148
71	90	1100	4600	3	3100	4781	3	3281	181	0	181
72	90	1100	4154	3	2654	4343	3	2843	189	0	189
73	90	1100	3046	2	2046	3722	3	2222	676	1	176
74	90	1100	4292	3	2792	4474	3	2974	182	0	182
75	90	1100	4324	3	2824	4497	3	2997	173	0	173
76	100	1200	4342	3	2842	4558	3	3058	216	0	216
77	100	1200	5822	4	3822	5964	4	3964	142	0	142
78	100	1200	4704	3	3204	4904	3	3404	200	0	200
79	100	1200	4178	3	2678	4403	3	2903	225	0	225
80	100	1200	4886	3	3386	5083	3	3583	197	0	197

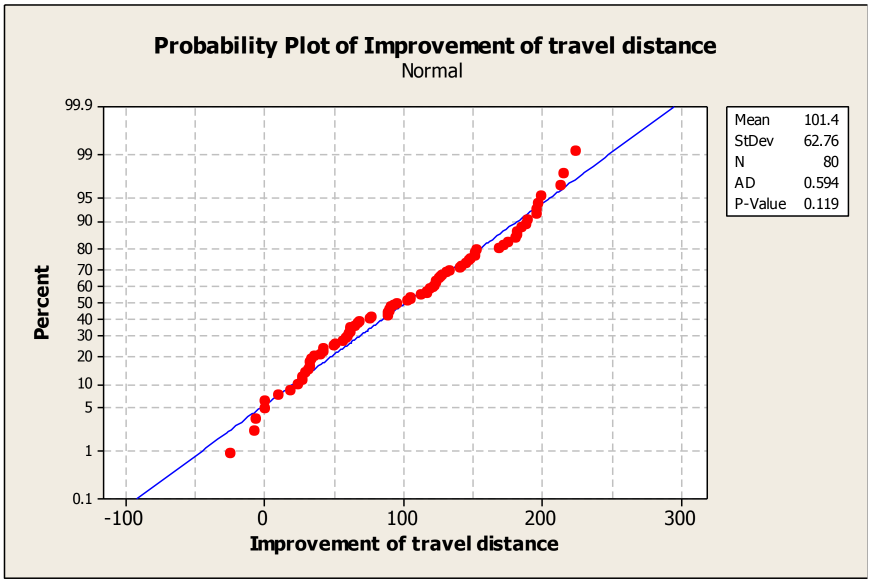

Figure A.2: Anderson-Darling Normality Test Result for Improvement of Travel Distance of LB. Minitab result.

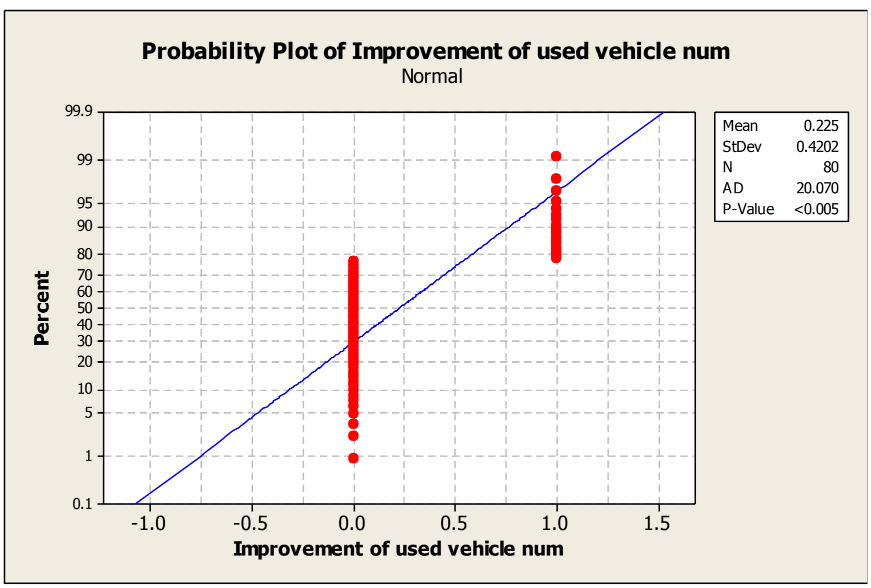

Figure A.3: Anderson-Darling Normality Test Result for Improvement of Used Vehicle Number of LB. Minitab result.

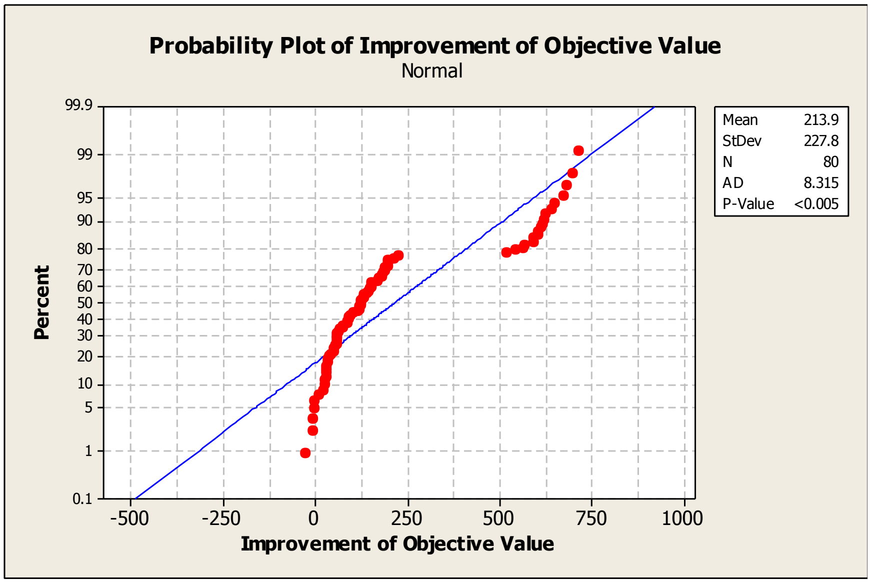

Figure A.4: Anderson-Darling Normality Test Result for Improvement of LB Value. Minitab result.

Wilcoxon Signed Rank Test: Improvement of LB value

Test of median = 0.000000 versus median > 0.000000

	N	N for Test	Wilcoxon Statistic	P	Estimated Median
diff of LB	80	80	3227.0	0.000	142.0

Figure A.5: Wilcoxon Signed Rank Test result for difference between LB values provided by new approach and old approach, improvement of LB value = (new approach value– old approach value), Minitab result

Wilcoxon Signed Rank Test: Improvement of Used Vehicle Number

Test of median = 0.000000 versus median > 0.000000

	N	N for Test	Wilcoxon Statistic	P	Estimated Median
diff of vehicle	80	18	171.0	0.000	0.000000000

Figure A.6: Wilcoxon Signed Rank Test result for difference between used vehicle number provided by new approach and old approach, improvement of used vehicle number = (new approach value– old approach value), Minitab result

Paired T-Test and CI: New LB Travel Distance, Old LB Travel Distance

Paired T for New LB Travel Distance - Old LB Travel Distance

	N	Mean	StDev	SE Mean
New LB Travel Distance	80	1295	1075	120
Old LB Travel Distance	80	1193	1023	114
Difference	80	101.41	62.76	7.02

95% CI for mean difference: (87.45, 115.38)
T-Test of mean difference = 0 (vs not = 0): T-Value = 14.45 P-Value = 0.000

Figure A.7: Paired T-Test result for difference between travel distance provided by new approach and old approach, improvement of used vehicle number = (new approach value– old approach value), Minitab result

Appendix B

Statistical Results: Running results for A3 analysis

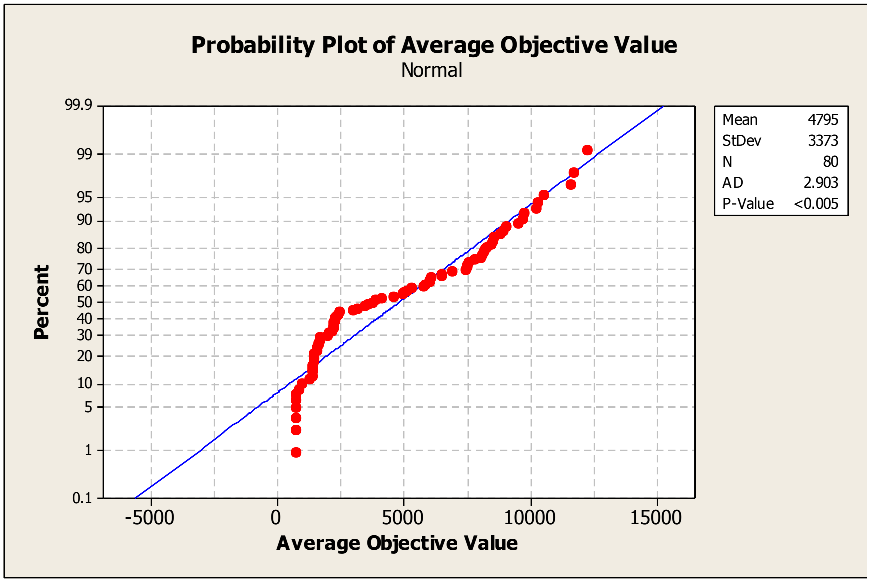

Figure B.1: Anderson-Darling Normality Test Result for Average Objective Value in A3 analysis. Minitab result.

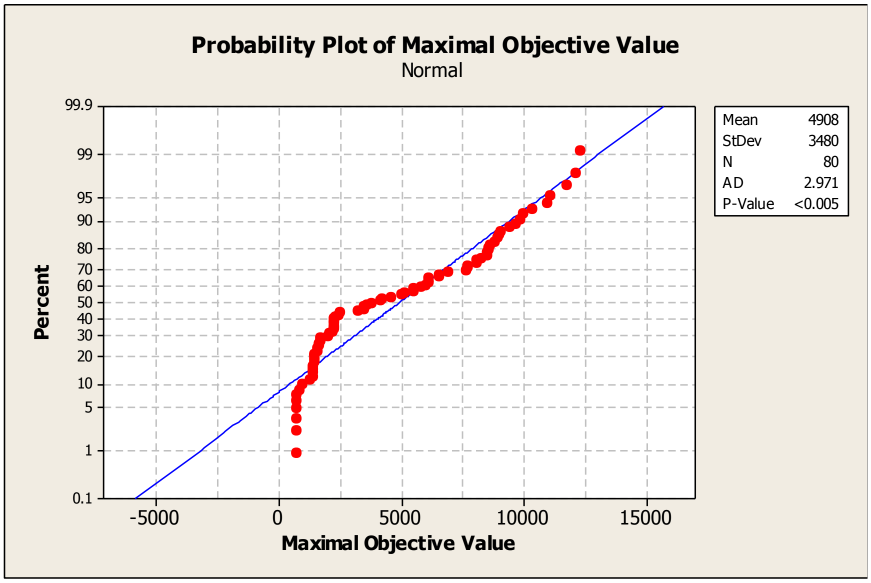

Figure B.2: Anderson-Darling Normality Test Result for Maximum Objective Value in A3 analysis. Minitab result.

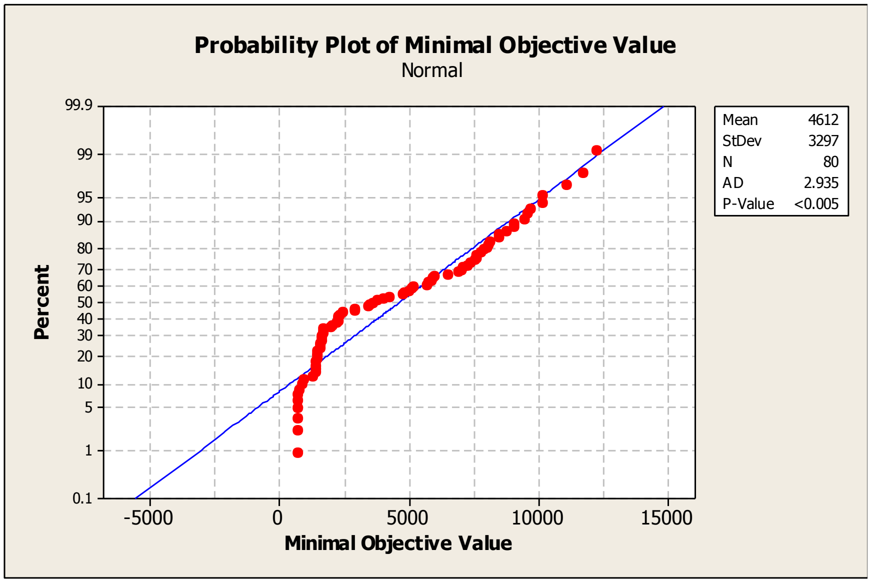

Figure B.3: Anderson-Darling Normality Test Result for Minimum Objective Value in A3 analysis. Minitab result.

Study: Minimal Objective Value versus Heuristic blocked by number of customers

Heuristics, Sum of the ranks

	ObjMin	replication
1	65.0	16
2	79.0	16
3	31.0	16
4	26.5	16
5	38.5	16

Friedman's Test
===============
Adjusted for ties
Value: 53.80064
Pvalue chisq : 5.793588e-11
F value : 79.12358
Pvalue F: 0

Alpha : 0.05
t-Student : 2.000298
LSD : 7.272029

Means with the same letter are not significantly different.

	GroupTreatment and Sum of the ranks	
a	2	79
b	1	65
c	5	38.5
d	3	31
d	4	26.5

Figure B.4: Freidman Test result for A3 Test1, Minimum Objective Value versus Heuristic blocked by number of customers, R result

Study: Maximum Objective Value versus Heuristic blocked by number of customers

Heuristics, Sum of the ranks

	ObjMax	replication
1	54.0	16
2	77.0	16
3	24.5	16
4	27.5	16
5	57.0	16

Friedman's Test
===============
Adjusted for ties
Value: 49.02857
Pvalue chisq : 5.75923e-10
F value : 49.12214
Pvalue F: 0

Alpha : 0.05
t-Student : 2.000298
LSD : 8.866984

Means with the same letter are not significantly different.

Group	Treatment	Sum of the ranks
a	2	77
b	5	57
b	1	54
c	4	27.5
c	3	24.5

Figure B.5: Freidman Test result for A3 Test2, Maximum Objective Value versus Heuristic blocked by number of customers, R result

Study: Average Objective Value versus Heuristic blocked by number of customers

Heuristics, Sum of the ranks

	ObjAvg	replication
1	59.0	16
2	79.0	16
3	26.5	16
4	24.5	16
5	51.0	16

Friedman's Test
===============
Adjusted for ties
Value: 52.80251
Pvalue chisq : 9.372414e-11
F value : 70.73348
Pvalue F: 0

Alpha : 0.05
t-Student : 2.000298
LSD : 7.716934

Means with the same letter are not significantly different.

	GroupTreatment and Sum of the ranks	
a	2	79
b	1	59
c	5	51
d	3	26.5
d	4	24.5

Figure B.6: Freidman Test result for A3 Test3, Average Objective Value versus Heuristic blocked by number of customers, R result

Appendix C

Statistical Results: Running results for A4 analysis

Table C.1: Changing of Minimum Objective Value of System Transportation Cost by changing 4 input parameters in A4 analysis

	Minimum Objective Value of System Transportation Cost			
% change From base	Fixed Cost	Variable Cost	Plant Production Rate	Product Lifetime
-90%	3800.1	2974.1	N/A	N/A
-80%	3990	3348.2	N/A	N/A
-70%	4272.4	3713.75	N/A	N/A
-60%	4630.6	4065.44	N/A	6197.9
-50%	4854.3	4470.5	N/A	6226.3
-40%	5141.1	4766.16	N/A	6150.4
-30%	5427.4	5159.24	6298.4	6280.9
-20%	5670.5	5606.8	6189.1	6114.1
-10%	5955.8	5861.97	6183.2	6233.8
0%	6181.5	6181.5	6181.5	6181.5
+10%	6538.3	6580.46	6284.3	6218.9
+20%	6731.5	6942.8	6246.3	6280.6
+30%	6959.2	7339.83	6263.7	6239.4
+40%	7208.8	7643.78	6257.6	6239.4
+50%	7555.7	8088.75	6171.2	6168.3
+60%	7853.3	8186.4	6264.3	6168.3
+70%	8090.3	8651.53	6166.7	6168.3
+80%	8382.5	9226.26	6270.2	6168.3
+90%	8702.7	9442.53	6219.9	6168.3

Table C.2: Changing of Maximum Objective Value of System Transportation Cost by changing 4 input parameters in A4 analysis

	Maximum Objective Value of System Transportation Cost			
% change From base	Fixed Cost	Variable Cost	Plant Production Rate	Product Lifetime
-90%	3955.8	3489.7	N/A	N/A
-80%	4225.8	3879.4	N/A	N/A
-70%	4758.3	4326.77	N/A	N/A
-60%	5224	4712.68	N/A	7159.6
-50%	5465.2	5076.8	N/A	7089.2
-40%	5759.6	5472.16	N/A	7146.9
-30%	6149.5	5914.56	7103.4	7059.7
-20%	6403.9	6262.88	7064.5	7131.2
-10%	6774	6640.52	7124.5	7018.8
0%	6994.9	6994.9	6994.9	6994.9
+10%	7374	7475.37	7058.3	7068.5
+20%	7719.5	7830.4	7058.3	6975.8
+30%	7961.6	8070.45	7131.4	7068.5
+40%	8307.9	8544.18	6880.5	7068.5
+50%	8648.3	9011.45	7124.5	7016
+60%	8957	9349.76	7029.5	7016
+70%	9237.5	9733.57	7067.7	7016
+80%	9548.6	10108	7303.1	7016
+90%	9961.8	10645.5	7303.1	7016

Table C.3: Changing of Average Objective Value of System Transportation Cost by changing 4 input parameters in A4 analysis

	Average Objective Value of System Transportation Cost			
% change From base	Fixed Cost	Variable Cost	Plant Production Rate	Product Lifetime
-90%	3841.022	3055.605	N/A	N/A
-80%	4149.755	3425.496	N/A	N/A
-70%	4508.543	3952.221	N/A	N/A
-60%	4812.355	4236.416	N/A	6854.761
-50%	5107.539	4718.659	N/A	6606.616
-40%	5378.904	5042.404	N/A	6519.384
-30%	5677.765	5486.113	6861.413	6576.931
-20%	5963.024	5887.608	6628.702	6530.865
-10%	6196.392	6103.14	6553.414	6513.308
0%	6463.014	6463.014	6463.014	6463.014
+10%	6859.822	6934.422	6642.331	6506.91
+20%	7198.527	7666.624	6565.635	6597.076
+30%	7319.653	7492.58	6570.998	6604.424
+40%	7781.951	7961.082	6428.951	6604.424
+50%	7968.494	8542.466	6496.482	6530.437
+60%	8330.484	8722.384	6475.071	6530.437
+70%	8535.394	9095.265	6486.571	6536.606
+80%	8886.241	9558.756	6499.588	6536.606
+90%	9251.329	9882.395	6504.478	6536.606

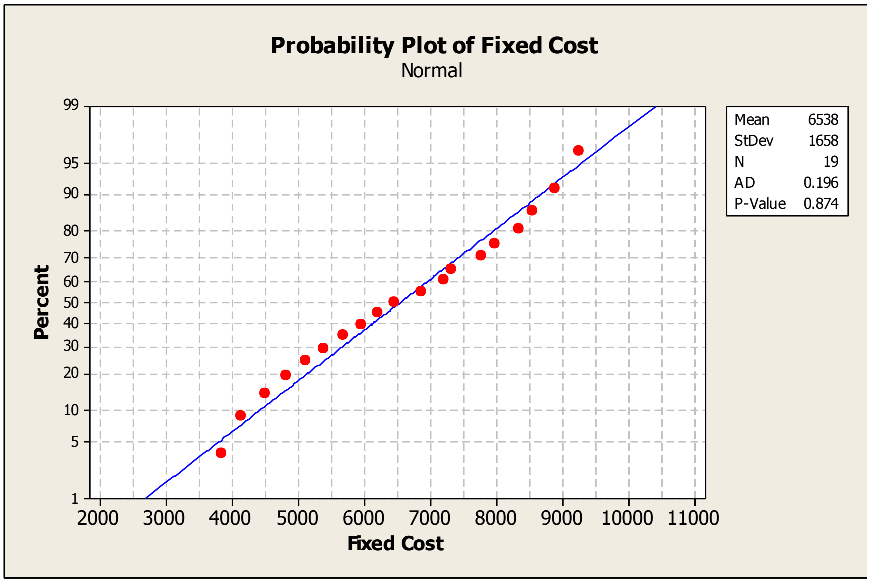

Figure C.4: Anderson-Darling Normality Test Result of Minimum Objective Value by changing fixed vehicle cost. Minitab result.

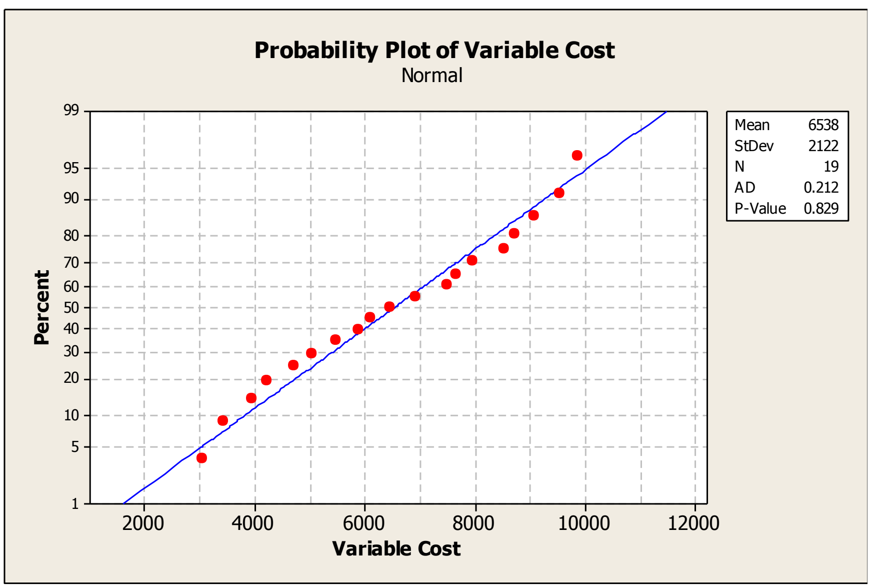

Figure C.5: Anderson-Darling Normality Test Result for Minimum Objective Value by changing variable vehicle cost. Minitab result.

Correlations: % change From base, Fixed Cost

Pearson correlation of % change From base and Fixed Cost = 1.000
P-Value = 0.000

Figure C.7: Pearson Correlations Test Result for Minimum Objective Value with fixed vehicle cost. Minitab result.

Correlations: % change From base, Variable Cost

Pearson correlation of % change From base and Variable Cost = 0.998
P-Value = 0.000

Figure C.8: Pearson Correlations Test Result for Minimum Objective Value with variable vehicle cost. Minitab result.

Appendix D

Statistical Results: Running Results for B2 Analysis

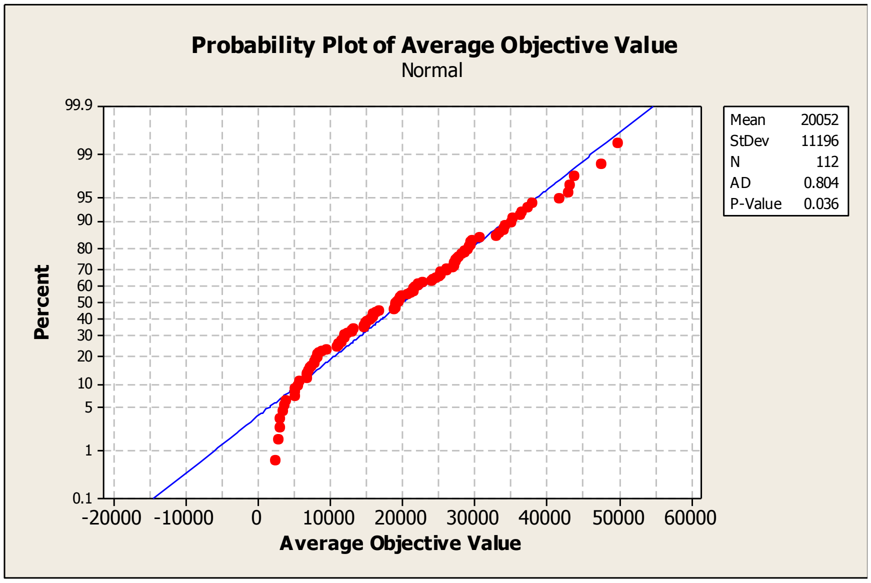

Figure D.1: Anderson-Darling Normality Test Result for Average Objective Value in B2 analysis. Minitab result.

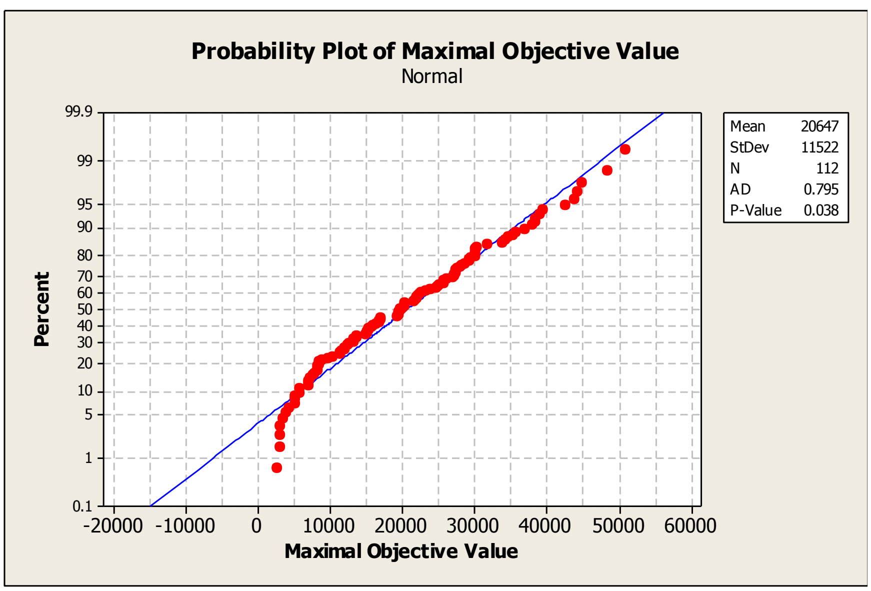

Figure D.2: Anderson-Darling Normality Test Result for Maximum Objective Value in B2 analysis. Minitab result.

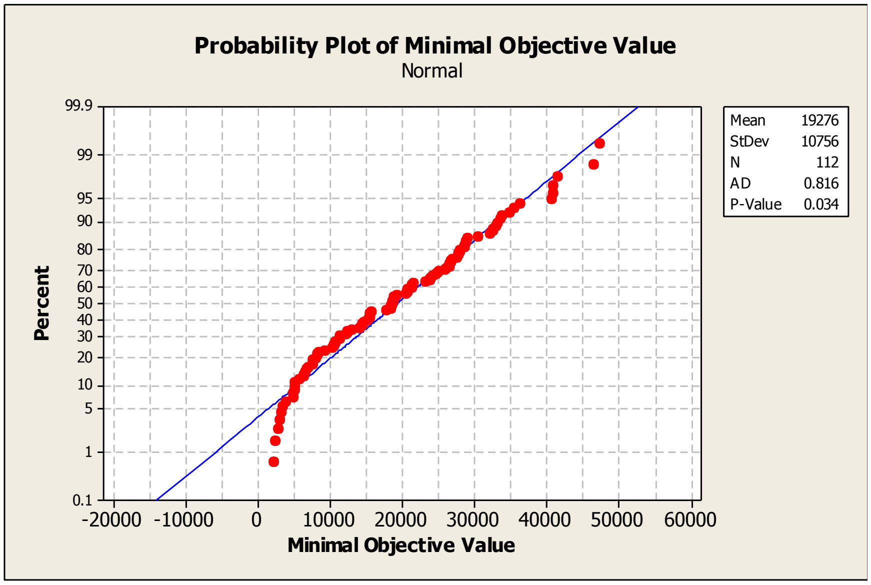

Figure D.3: Anderson-Darling Normality Test Result for Minimum Objective Value in B2 analysis. Minitab result.

Study: Minimum Objective Value versus Heuristic blocked by number of customers

Heuristics, Sum of the ranks

	ObjMin	replication
M1	41	16
M2	70	16
M3	59	16
M4	47	16
M5	108	16
M6	99	16
SA	24	16

Friedman's Test
===============
Adjusted for ties
Value: 75.53571
Pvalue chisq : 2.975398e-14
F value : 55.36649
Pvalue F: 0

Alpha : 0.05
t-Student : 1.986675
LSD : 11.57663

Means with the same letter are not significantly different.

Group	Treatment and	Sum of the ranks
a	M5	108
a	M6	99
b	M2	70
b	M3	59
c	M4	47
c	M1	41
d	SA	24

Figure D.4: Freidman Test result for B2 analysis Test 01, Minimum Objective Value versus Heuristic blocked by number of customers; MINITAB output

Study: Maximum Objective Value versus Heuristic blocked by number of customers

Heuristics, Sum of the ranks

	ObjMax	replication
M1	41.5	16
M2	64.0	16
M3	47.5	16
M4	54.0	16
M5	107.0	16
M6	101.0	16
SA	33.0	16

Friedman's Test
===============
Adjusted for ties
Value: 67.81006
Pvalue chisq : 1.149081e-12
F value : 36.08205
Pvalue F: 0

Alpha : 0.05
t-Student : 1.986675
LSD : 13.57965

Means with the same letter are not significantly different.

Group	Treatment and	Sum of the ranks
a	M5	107
a	M6	101
b	M2	64
bc	M4	54
c	M3	47.5
cd	M1	41.5
d	SA	33

Figure D.5: Freidman Test result for B2 analysis Test 02, Maximum Objective Value versus Heuristic blocked by number of customers; MINITAB output

Study: Average Objective Value versus Heuristic blocked by number of customers

Heuristics, Sum of the ranks

	ObjAvg	replication
M1	38	16
M2	68	16
M3	56	16
M4	51	16
M5	108	16
M6	100	16
SA	27	16

Friedman's Test
===============
Adjusted for ties
Value: 74.00893
Pvalue chisq : 6.139533e-14
F value : 50.48112
Pvalue F: 0

Alpha : 0.05
t-Student : 1.986675
LSD : 12.00071

Means with the same letter are not significantly different.

Group	Treatment and	Sum of the ranks
a	M5	108
a	M6	100
b	M2	68
bc	M3	56
c	M4	51
d	M1	38
d	SA	27

Figure D.6: Freidman Test result for B2 analysis Test 03, Average Objective Value versus Heuristic blocked by number of customers; MINITAB output

Appendix E

Statistical Results: Running Result for B3 Analysis

Table E.1: Running result of B2 analysis.

% change from base	Minimum Objective Value of System Transportation Cost				
	Fixed Cost	Variable Cost	Product Lifetime	Plant 1 Production Rate	Plant 2 Production Rate
-90%	13630.8	9309.3	N/A	N/A	N/A
-80%	14706.6	11109.4	N/A	N/A	N/A
-70%	15417.6	11933.7	N/A	N/A	N/A
-60%	15942.8	13730.8	N/A	N/A	N/A
-50%	17026.5	15003.5	N/A	N/A	21146
-40%	18143.5	16372.3	N/A	21522.9	21491
-30%	18599.2	17589.4	21998	20939	21521.8
-20%	19801.5	18398.6	21569	20989.3	21553.7
-10%	20583.9	20159.6	20906.8	21228.9	20977.2
0%	20981.8	21520.7	21002.5	21152.6	21623.8
+10%	21025.1	22741.3	20744.8	21435.6	20724
+20%	22470.8	23655.2	20969.9	21459	21349.6
+30%	23968.2	25295.4	21239	21427	20991
+40%	24863	26375	20640.3	21460.9	20877.4
+50%	24925	27799.4	20944.2	21052	20927.3
+60%	25719.6	29161.1	20904.6	21154.1	20919.4
+70%	27414.3	29895	20884.9	21361.8	20831
+80%	27160.1	31208.2	20884.8	20622.9	21585.4
+90%	27004.1	32549.5	20899.1	21182.8	20796

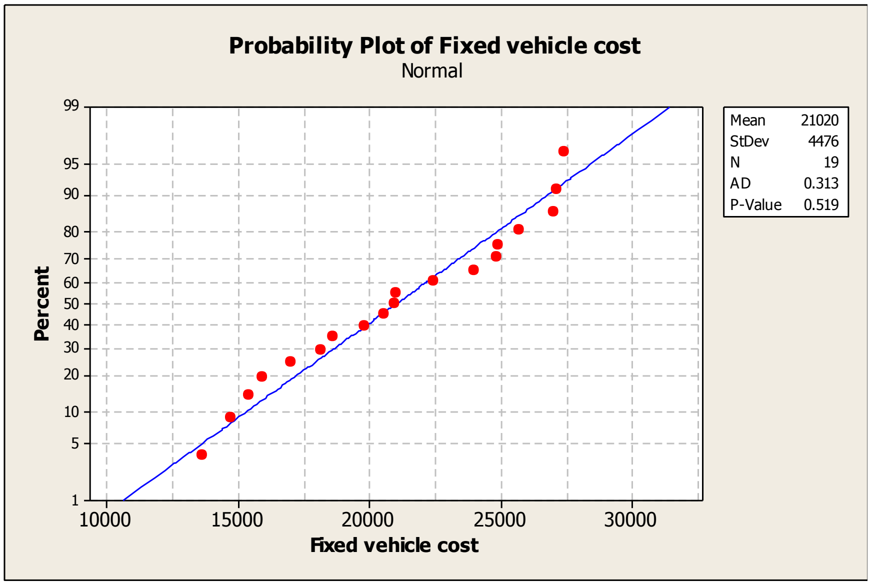

Figure E.2: Anderson-Darling Normality Test Result for fixed vehicle cost in B3 analysis. Minitab result.

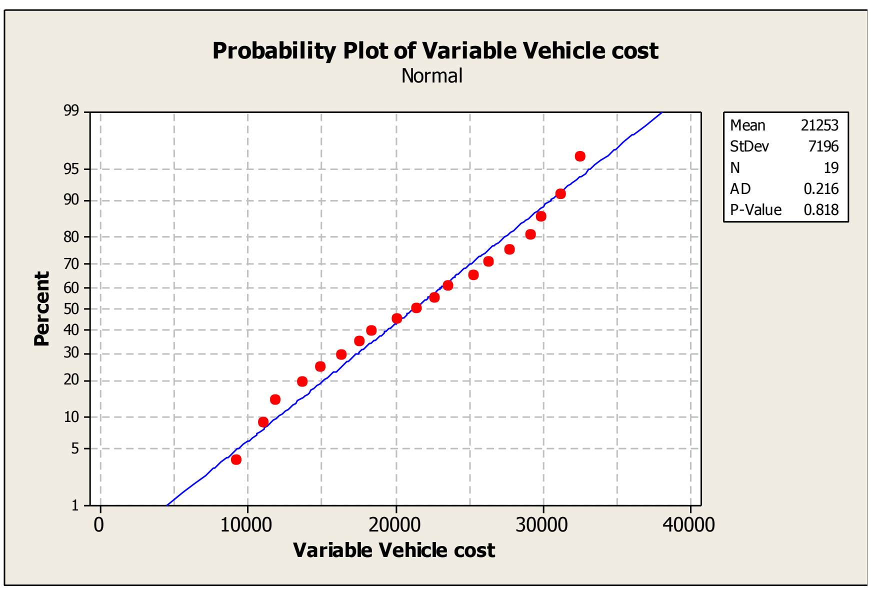

Figure E.3: Anderson-Darling Normality Test Result for variable vehicle cost in B3 analysis. Minitab result.

Correlations: Fixed vehicle cost, Changed Percentage

Pearson correlation of Fixed vehicle cost and Changed Percentage = 0.994
P-Value = 0.000

Figure E.4: Pearson Correlations Test Result for Minimum Objective Value with fixed vehicle cost. Minitab result.

Correlations: Variable Vehicle cost, Changed Percentage

Pearson correlation of Variable Vehicle cost and Changed Percentage = 0.999
P-Value = 0.000

Figure E.5: Pearson Correlations Test Result for Minimum Objective Value with variable vehicle cost. Minitab result.

REFERENCES

Armstrong R, Gao S, Lei L. A zero-inventory production and distribution problem with a fixed customer sequence. Annals of Operations Research 2008;159:395-414.

Bilgen B, Ozkarahan I. Strategic tactical and operational production-distribution models: a review. International Journal of Technology Management 2004;28:151-71.

Bodin L, Golden B. Classification in vehicle routing and scheduling. Networks 1981;11:97-108.

Bodin LD. A taxonomic structure for vehicle routing and scheduling problems. Computers &Urban Society 1975;1:11-29.

Brandão JCS, Mercer A. The multi-trip vehicle routing problem. The Journal of the Operational Research Society 1998;49:799-805.

Chandra P, Fisher ML. Coordination of production and distribution planning. European Journal of Operational Research 1994;72:503-17.

Chang Y-C, Lee C-Y. Machine scheduling with job delivery coordination. European Journal of Operational Research 2004;158:470-87.

Chen H-K, Hsueh C-F, Chang M-S. Production scheduling and vehicle routing with time windows for perishable food products. Computers & Operations Research 2009;36:2311-19.

Chen Z-L. Integrated production and distribution operations: taxonomy, models, and review. In Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business: Kluwer Academic Publishers, 2004.

Chen Z-L. Integrated production and outbound distribution scheduling: review and extensions. Operations Research 2009;58:130-148.

Chen ZL, Vairaktarakis GL. Integrated scheduling of production and distribution operations. Management Science 2005;51:614-28.

Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 1964;12:568-81.

Crevier B, Cordeau J-F, Laporte G. The multi-depot vehicle routing problem with inter-depot routes. European Journal of Operational Research 2007;176:756-73.

Dantzig G, Fulkerson R, Johnson S. Solution of a large-scale traveling-salesman problem. Journal of the Operations Research Society of America 1954;2:393-410.

Devapriya, P, Optimal Fleet Size of an Integrated Production and Distribution Scheduling Problem for a Single Perishable Product, PHD Dissertation, 2008.

Eksioglu B, Vural AV, Reisman A. The vehicle routing problem: A taxonomic review. Computers & Industrial Engineering 2009;57:1472-83.

Erenguc SS, Simpson NC, Vakharia AJ. Integrated production/distribution planning in supply chains: An invited review. European Journal of Operational Research 1999;115:219-36.

Fumero F, Vercellis C. Synchronized development of production, inventory, and distribution schedules. Transportation Science 1999;33:330-40.

Garcia JM, Lozano S, Canca D. Coordinated scheduling of production and delivery from multiple plants. Robotics and Computer-Integrated Manufacturing 2004;20:191-98.

Geismar HN, Laporte G, Lei L, Sriskandarajah C. The integrated production and transportation scheduling problem for a product with a short lifespan. Informs Journal on Computing 2008;20:21-33.

Ghare PM, Schrader, G. F. A model for exponentially decaying inventory. Journal of Industrial Engineering 1963;14:238-43.

Hall LA, Shmoys DB. Jackson's rule for single-machine scheduling: making a good heuristic better. Mathematics of Operations Research 1992;17:22-35.

Ho W, Ho GTS, Ji P, Lau HCW. A hybrid genetic algorithm for the multi-depot vehicle routing problem. Engineering Applications of Artificial Intelligence 2008;21:548-57.

Hochbaum DS, Hong SP. On the complexity of the production-transportation problem. Siam Journal on Optimization 1996;6:250-64.

Horn WA. Minimizing average flow time with parallel machines. Operations Research 1973;21:846-47.

Hunter AP, Van Buer MG. The newspaper production/distribution problem. Journal of Business Logistics 1996;17:85-107.

Ishii K, Takahashi K, Muramatsu R. Integrated production, inventory and distribution systems. International Journal of Production Research 1987;26:473 - 82.

Jayaraman V, Pirkul H. Planning and coordination of production and distribution facilities for multiple commodities. European Journal of Operational Research 2001;133:394-408.

Jensen MT, Khouja M. An optimal polynomial time algorithm for the common cycle economic lot and delivery scheduling problem. European Journal of Operational Research 2004;156:305-11.

Jeon G, Leep HR, Shim JY. A vehicle routing problem solved by using a hybrid genetic algorithm. Computers & Industrial Engineering 2007;53:680-92.

Kamke, E. Theory of Sets. New York: Dover, p. 6, 1950.

Kuno T, Utsunomiya T. A Lagrangian Based Branch-and-Bound Algorithm for Production-transportation Problems. Journal of Global Optimization 2000;18:59-73.

Lee C-Y, Chen Z-L. Machine scheduling with transportation considerations. Journal of Scheduling 2001;4:3-24.

Li K, Ganesan VK, Sivakumar AI. Scheduling of single stage assembly with air transportation in a consumer electronic supply chain. Computers & Industrial Engineering. 2006;51:264-78.

Li K, Sivakumar AI, Ganesan VK. Complexities and algorithms for synchronized scheduling of parallel machine assembly and air transportation in consumer electronics supply chain. European Journal of Operational Research 2008;187:442-55.

Li KPG, V.K. and A.I.Sivakumar, . Synchronized scheduling of assembly and multi-destination air transportation in a consumer electronics supply chain. International Journal of Production Research 2005;43:2671-85.

Li KS, Appa Iyer; Mathirajan, M.; Ganesan, Viswanath Kumar. Solution methodology for synchronizing assembly manufacturing and air transportation of consumer electronics supply chain. International Journal of Business 2004;9:361-80.

Liu R, Jiang Z, Fung RYK, Chen F, Liu X. Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. Computers & Operations Research 2010;37:950-59.

Min H, Jayaraman V, Srivastava R. Combined location-routing problems: a synthesis and future research directions. European Journal of Operational Research 1998;108:1-15.

Mirabi M, Fatemi Ghomi SMT, Jolai F. Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem. Robotics and Computer-Integrated Manufacturing 2010;26:564-69.

Nahmias S. Perishable inventory theory: a review. Operations Research 1982;30:680-708.

Olivera A, Viera O. Adaptive memory programming for the vehicle routing problem with multiple trips. Computers & Operations Research 2007;34:28-47.

Owen SH, Daskin MS. Strategic facility location: a review. European Journal of Operational Research 1998;111:423-47.

Palekar US, Karwan MH, Zionts S. A branch-and-bound method for the fixed charge transportation problem. Management Science 1990;36:1092-105.

Park YB. An integrated approach for production and distribution planning in supply chain management. International Journal of Production Research 2005;43:1205-24.

Petch RJ, Salhi S. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. Discrete Applied Mathematics 2003;133:69-92.

Renaud J, Laporte G, Boctor FF. A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research. 1996;23:229-35.

Ronen D. Perspectives on practical aspects of truck routing and scheduling. European Journal of Operational Research 1988;35:137-45.

Salhi S. The integration of routing into the location-allocation and vehicle composition problems. University of Lancaster, 1987.

Salhi S, Sari M. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. European Journal of Operational Research 1997;103:95-112.

Sarmiento AM, Nagi R. A review of integrated analysis of production-distribution systems. Iie Transactions 1999;31:1061-74.

Strusevich VA. A heuristic for the two-machine open-shop scheduling problem with transportation times. Discrete Applied Mathematics 1999;93:287-304.

Sumichrast RT, Markham IS. A heuristic and lower bound for a multi-depot routing problem. Computers & Operations Research. 1995;22:1047-56.

Taillard ED, Laporte G, Gendreau M. Vehicle routeing with multiple use of vehicles. The Journal of the Operational Research Society 1996;47:1065-70.

Thomas DJ, Griffin PM. Coordinated supply chain management. European Journal of Operational Research 1996;94:1-15.

Toth P, Vigo D. Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Applied Mathematics 2002;123:487-512.

Tuy H, Dinh Dan N, Ghannadan S. Strongly polynomial time algorithms for certain concave minimization problems on networks. Operations Research Letters 1993;14:99-109.

Tuy H, Ghannadan S, Migdalas A, Värbrand P. A strongly polynomial algorithm for a concave production-transportation problem with a fixed number of nonlinear variables. Mathematical Programming. 1996;72:229-58.

Vidal CJ, Goetschalckx M. Strategic production-distribution models: A critical review with emphasis on global supply chain models. European Journal of Operational Research 1997;98:1-18.

Woeginger GJ. Heuristics for parallel machine scheduling with delivery times. Acta Informatica. 1994;31:503-12.

Woeginger GJ. A polynomial-time approximation scheme for single-machine sequencing with delivery times and sequence-independent batch set-up times. Journal of Scheduling 1998;1:79-87.

Youssef MA, Mahmoud MM. An iterative procedure for solving the uncapacitated production-distribution problem under concave cost function. International Journal of Operations & Production Management 1996;16:18-27.

Zdrzalka S. Preemptive scheduling with release dates, delivery times and sequence independent setup times. European Journal of Operational Research 1994;76:60-71.

Zegordi SH, Nia MAB. Integrating production and transportation scheduling in a two-stage supply chain considering order assignment. International Journal of Advanced Manufacturing Technology 2009;44:928-39.