2002

Operational strategies for cross docking systems

Wooyeon Yu
Iowa State University

Operational strategies for cross docking systems

by

Wooyeon Yu

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Pius J. Egbelu, Major Professor
Douglas Gemmill
Sarah Ryan
Timothy Van Voorhis
Yoshinori Suzuki

Iowa State University

Ames, Iowa

2002

UMI Number: 3051505

UMI®

UMI Microform 3051505

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Wooyeon Yu

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

DEDICATION

To my parents, wife and son

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Cross docking is a warehouse management concept in which items delivered to a warehouse by delivery trucks are immediately sorted out and reorganized based on customer demands and are routed and loaded into shipping trucks for delivery to customers without actually being held in inventory in the warehouse. If any item is to be held in storage, it is only for a brief period of time that is generally less than twenty-four hours. This way, the turnaround times for customer orders, inventory management cost, and warehouse space requirements are reduced. Because accuracy in material management is required in such operations, a cross docking operation is heavily dependent on accurate flow of information.

Depending on the facility and operating conditions or strategies employed, it is possible to generate various cross docking scenarios or models. In this research, thirty-two different models are identified based on the number of docks available at the site, the dock holding pattern for trucks, and the existence of temporary storage. Of the thirty-two models identified, this research is focused on three. All three models assume there is a separate truck receiving dock and a separate truck shipping dock. It is also assumed that the items contained in a receiving truck and the items needed for a shipping truck are known in advance. Furthermore, the study is restricted to scenarios where there is only one shipping dock and only one receiving dock at the warehouse.

In the first model of the cross docking problem studied, it is assumed there is temporary storage in front of the shipping dock. If a product that arrives at the shipping dock does not need to be loaded into the shipping truck currently at the dock, the product can be stored in a temporary storage area until the appropriate shipping truck comes into the shipping dock. In this model, both the receiving and the shipping trucks must stay in docks until they finish their unloading or loading tasks once they come into docks. Therefore, a receiving truck cannot leave the receiving dock until all of its products are unloaded onto the receiving dock. Similarly, a shipping truck cannot leave the shipping dock until all of its needed products are loaded.

In the second model investigated, it is assumed that no temporary storage exists in the warehouse. However, both the receiving truck and the shipping truck can move in and out of

the dock during their tasks. Therefore, it is possible that a receiving truck unloads some of its products on the receiving dock, moves out, waits and goes into the receiving dock again to unload its remaining products. This operating pattern can be similarly applied to the shipping truck. However, since there is no temporary storage space available, the conveyor operating from the receiving dock to the shipping dock may need to stop if the shipping truck is not ready when a product arrives at the shipping dock.

In the third and final model investigated, it is assumed that there is temporary storage in front of the shipping dock and that both the receiving trucks and the shipping trucks can intermittently move in and out of the dock during the time intervals between their task execution. After a receiving truck unloads some of its products for a certain shipping truck, one of two choices can be made; either more products are unloaded from the current receiving truck and sent to the temporary storage, or the current receiving truck is moved out from the receiving dock and another receiving truck is sent to the receiving dock to unload its products. Thereafter, the earlier truck is rescheduled at the dock at a later time to continue its unloading process. A similar operation plan is applicable to the shipping trucks as well.

One of the objectives for cross docking systems is how well the trucks can be scheduled at the dock and how the items in receiving trucks can be allocated to the shipping trucks to optimize on some measure of system performance. In all the cross docking scenarios studied, the research objective is to find the best truck spotting sequence for both receiving and shipping trucks to minimize total operation time (i.e., the makespan) or to maximize the throughput of the cross docking system. The product routing and the spotting sequences of the receiving and shipping trucks are all determined simultaneously.

Several solutions approaches are employed in modeling and solving the problems. The approaches are also adapted to the models. The solution approaches employed include mixed integer programming, branch and bound, search algorithm, complete enumeration, and heuristics. The complete enumeration and the mixed integer programming approaches were used as the basis to generate optimal solutions with which the performances of the heuristic and search algorithms could be benchmarked. From the results of the test problems, the heuristics and the search algorithms produced very good and competitive solutions when compared with the solutions obtained through the exact procedure and the complete enumeration. The results obtained also indicate that the performance of the cross docking system is dependent on the strategies employed in operating the system. As expected, the less restrictive the system operates, the better the system performs, with model three generally producing better results than the two other models for the same set of operating parameters. Another finding is that truck change time at the docks influences the sequencing of the trucks and the solutions obtained for any given scenario. As the truck change time changes, the sequencing of the shipping and the receiving trucks also may change.

The research is important for a number of reasons. First, the problem addressed is very real and is one that is faced daily in supply chain networks. Second, the work represents, perhaps, the first full technical study reported on the subject matter. Third, the innovative solution approaches developed will provide a basis for operating more efficient cross docking warehouses. Fourth, the dissertation identifies a whole new area of research in supply chain engineering. Finally, when designed and analyzed in the way undertaken in this research, warehouse operators can expect to save millions of dollars annually in their operations. More importantly, the reduced warehousing cost will be achieved while reducing the leadtime on customer orders.

CHAPTER 1. INTRODUCTION

1.1 A Warehouse or Distribution Center

A typical warehouse is a dynamic and intelligent distribution center in which products and packages are processed in real time and moved in and out on schedule. A dynamic and intelligent warehouse is also a place where all distribution and logistic functions are tied together and where inventory storage is minimal. The input and output are also precisely regulated and streamlined in an intelligent manner.

In today's distribution environment, the pressure is on to make the operations more efficient. Companies are cutting costs by reducing inventory at every step of the operation, including distribution. Customers are demanding better service, which translate into more accurate and timely shipments. Instead of waiting a week to get a product, most customers expect to receive a delivery in one or two days. In most manufacturing environments, it is difficult to ship directly from the manufacturers to the customer. Therefore, a certain type of intermediate points is necessary to connect between manufacturers and customers. One type of intermediate point in a supply chain system is the distribution center.

Operations of the distribution center consist of five basic functions: receiving, sorting, storing, picking and shipping. If the way these five elements cooperate is improved, costs can be reduced and productivity can be improved. However, the best way to reduce cost and improve efficiency is not by simply improving a function but by eliminating it if feasible. Cross docking has the potential of eliminating storage and picking, the two most expensive warehousing operations. Cross docking is a method of distribution management that helps companies better control their distribution operations.

1.2 Cross Docking System

Cross docking is a material handling and distribution concept in which items move directly from receiving dock to shipping dock, without being stored in a warehouse or distribution center. In a typical cross docking system, the primary objective is to eliminate storage and excessive material handling.

Through the years, cross docking has had many names. It has been called "expediting" orders or "opportunistic" shipping. The goal has always been the same: reduce material handling by moving goods directly to the end user, bypassing storage (Witt, 1998).

Modern technology makes cross docking feasible. Without computers, automatic-identification technology, and other kinds of materials handling equipment, companies would not be able to transfer huge quantities of boxes or pallets rapidly enough from one truck to another (Cooke, 1996).

Figures 1 and 2 show the flow of material in a typical cross docking operation (Rohrer, 1995). As shown in Figures 1 and 2, the cross docking system generally operates as follows:

1. Products (packages, boxes, cartons, etc.) arrive on the receiving docks.
2. Products are scanned and verified at the receiving docks. In some cross docking systems products are also weighed, sized and labeled at the receiving dock.
3. Products are placed on the sortation systems, which sort by destinations.
4. Products are processed to the proper location on the shipping docks.

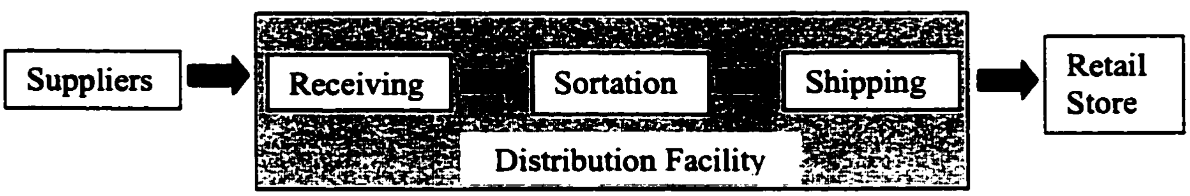


Figure 1. Typical Cross Docking Flow (Rohrer, 1995)

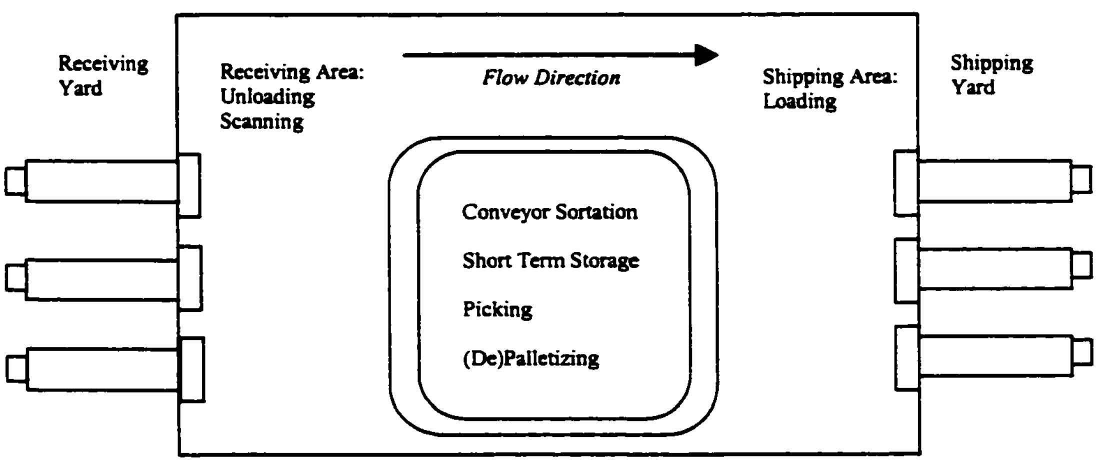


Figure 2. Typical Cross Docking Distribution Facility (Rohrer, 1995)

Cross docking can apply to both manufacturing and distribution functions (Schwind, 1996). All warehouses or distribution centers do some cross docking. In a manufacturing plant, the finished product ordinarily goes from the packaging point to storage. With cross docking, the product travels directly from packaging to shipping, sometimes directly into the waiting trucks. In a distribution center, cross docking can involve most or all of the material that arrive at the receiving dock. Package handling services are a good example. At package delivery companies, such as the U.S. Postal Service, United Parcel Service, Federal Express, and many others, everything they receive must be broken down, sorted and shipped out as soon as possible. No inventory is held for even 24 hours.

In general, it seems that cross docking works best for companies either distributing large volumes of merchandise or serving a large number of stores. Cross docking systems handle a high volume of items in a short amount of time. The advantages of cross docking systems include increased inventory turn, thus reduced inventory, increased customer responsiveness, and better control of the distribution operation.

1.3 Implementation of Cross Docking Systems

Cross docking might look simple, but there are a number of unseen critical activities. Here are a few requirements for successful cross docking (Schaffer, 1998).

- Partnering with other members of the distribution chain;
- Communication among all members of a supply chain;
- Absolute confidence in the quality and availability of the product;
- Communication and control within the cross docking operation;
- Personnel, equipment and facilities;
- Operational management.

Many operations have all the physical elements in place required for cross docking. In these operations, there is a temptation to implement cross docking without developing a formal program. Since cross docking requires that many internal and external functions should work closely together, attempting to implement it without a formal program is the path to failure. In order to successfully implement cross docking, a formal program must be set up to address each of the above categories. In addition, cross-docking strategies must be a try-and-adjust procedure until a working technique evolves.

1.4 Research Objectives

In many ways, operational management is the least considered but most important part of implementing cross docking. With all the planning, partnering, addition of equipment and systems and changes in manpower, cross docking still requires a high level of operational execution to work. For example, no matter how well the cross docking system is designed, it is still necessary for someone to coordinate the receiving and shipping trucks to the appropriate docks in the appropriate sequences. Improper sequencing of receiving and shipping trucks at the docks increases operation completion time.

Unfortunately, the need for operational management is often neglected and the function is added to the work of an already busy first-line supervisor. To prevent the lack of operational management from becoming a barrier to successful cross docking implementation, operational management should be developed simultaneously.

In this research, one area of operational management is developed. The operational management area developed in this research is the spotting sequences of the receiving and shipping trucks to the receiving and shipping docks in order to minimize total operation time or in order to maximize the throughput of the cross docking system. The product routing between the receiving and shipping trucks is also decided simultaneously as well as the spotting sequences of the receiving and shipping trucks.

1.5 Description of Basic Cross Docking Models and Solution Approaches Used

In this research, the following cross docking system is considered. The cross docking system of this study is operated as follows:

1. Receiving trucks arrive at the receiving docks and unload products onto the receiving dock.
2. Products move from the receiving dock to the shipping dock on a conveyor.
3. Shipping trucks load products from shipping docks and leave shipping docks.

The cross docking system in this research does not consider the operation inside the warehouse or distribution center, such as scanning and sorting operations, etc.

Depending on the facilities, operating conditions or strategies employed, it is possible to generate various cross docking models. In this research, thirty-two cross docking models are identified based on the number of docks available at the site, the dock holding pattern for trucks, and the existence of temporary storage. The detailed explanations for the thirty-two models are presented in Section 3.1.1. Among the thirty-two models, three specific models of the cross docking systems are considered in this research. The detailed explanations for the three specific models are presented in Section 3.2.

For each cross docking model studied in this research, both the mathematical model and the associated efficient heuristic methods for sequencing both receiving and shipping trucks are developed to minimize the total operation time or to maximize the throughput of the cross docking system. The allocations of the products from receiving trucks to shipping trucks are decided simultaneously as well as the spotting sequence of the receiving and shipping trucks. In each case, special structures of the model are exploited in developing the heuristic methods.

1.6 Justification of the Research

Modern technology makes cross docking feasible. Cross docking systems have two characteristics: hardware and software. Since cross docking systems are highly automated, it is necessary to have appropriate equipment. Meanwhile, software keeps the cross docking system running smoothly. This includes loading and spotting algorithms, product tracking systems, and information transfer with vendors.

Both hardware and software are important for cross docking success. Hardware such as material handling devices, sortation systems and computers has been continuously developed. Therefore, most of the required hardware for a cross docking system is available today. Meanwhile, software is relatively less developed, though it is as important as hardware to cross docking success. For example, the function of operational management is added to the work of an already busy first-line supervisor. To prevent the lack of operational management from becoming a barrier to successful cross docking implementation, the appropriate software should be implemented simultaneously.

Although operational management plays important role in cross docking success, there has been very few publications about the operational management of the cross docking systems. This lack of research dealing with the development of decision tools for cross docking systems motivated the work undertaken in this study. The algorithm developed in this research will provide the basis for a systematic management of cross docking operation and the development of next generation of control software for warehouse operations.

1.7 Organization of the Dissertation

This dissertation has been organized as follows: Chapter 2 provides the review of previous literature on cross docking systems. In Chapter 3, the general description of the cross docking system and brief explanation of the three models addressed in this dissertation are presented. The three models are referred throughout the dissertation as *Case 1 model*, *Case 2 model* and *Case 3 model* respectively. Chapter 4 presents the modeling and the solution approaches used for *Case 1 model*. In Chapter 5, the modeling and the solution approaches used for *Case 2 model* are presented. Chapter 6 presents the modeling and the solution approaches used for *Case 3 model*. The conclusions and suggestions for future research are presented in Chapter 7.

CHAPTER 2. LITERATURE REVIEW

In many warehouses or distribution centers, cross docking systems are implemented successfully. Nevertheless, relatively few research papers or articles are published on cross docking systems. As a result, no systematic or generally accepted approach for planning cross docking operation has emerged.

One of the first articles on cross docking systems was written by Wurz (1994). In his article, Wurz wrote "the warehouse of the future will be a dynamic and intelligent distribution center. The concept of a dynamic and intelligent distribution center is already manifested in cross docking operation." He argued that most of the technology for cross docking is available today. Automatic Identification (Auto. ID) technology is mainly discussed in his article among a variety of technologies for cross docking systems.

The first and only technical paper found on cross docking systems was presented by Rohrer (1995). He discussed modeling methods and issues as they apply to cross docking systems. His paper also described how simulation helps ensure success in cross docking systems by determining optimal hardware configuration and software control, as well as establishing failure strategies before cross docking problems are encountered. His paper is oriented toward simulation practitioners who need to model cross docking systems, as well as distribution managers who evaluate cross docking. He addressed that a simulation model for cross docking should retain as much of the details as possible while still allowing for timely completion of the model. Though he discussed modeling methods and issues for cross docking systems, there is no implementation shown in his paper.

When a cross docking system is considered, there are several points to remember. Schwind (1995) discussed considerations for cross docking systems. He mentioned that "there are two levels of cross docking: 1) physical handling equipment and strategy, and 2) information system strategy." The impacts of cross docking on material handling, electronic data interchange, automatic identification and dock management are also discussed in his article. He also addressed the issues that "simulation is important to cross docking not only when new equipment is added to a system, but also when new or enhanced information systems are added. There are always alternatives that require evaluation, and simulation can do this work with little risk."

A few articles in trade magazines report the successful implementation of cross docking systems. Forger (1995) discussed the success of cross docking operation of Chicago Area Consolidation Hub (CACH) of UPS. He explained how cross docking applied to CACH. According to this article, "even during early startup days, it only takes 15 minutes for a package to travel an average distance of one mile on a series of conveyors from one of 122 receiving docks to any of 1050 shipping docks." Total cost of CACH was $315 million. The expected daily throughput of cross docking is 2 million packages.

Some considerations on the equipment and procedures for cross docking success are presented in Schwind's article. He discussed the underlying principles, dock management factors, storage in the meantime, packaging, and design for cross docking. He addressed the layout and design of receiving and shipping docks as major parts of any cross docking system. "The smoothness with which trucks arrive, are accessed, unloaded/loaded and depart greatly influences cross docking success. Most of this activity takes place on the dock. Dock design to enable smooth cross docking involves many factors such as space, equipment, manpower, carrier management, dock management and information management. For a successful implementation of cross docking, the above issues should be fully considered before cross docking is implemented" (Schwind, 1996).

Cooke (1996) explained the necessary equipment such as the bar code, material handling and sortation equipment for a cross docking operation. He argued the right equipment is critical to the success of a cross docking operation. According to his article, "the necessary equipment for a retail cross docking operation can cost at least $500,000." He presented what is essential to set up a cross docking facility or convert an existing facility to accommodate a cross docking operation. In his article, some vendors that provide equipment needed for cross docking operation are also identified.

The concepts of cross docking are found in many articles. Among these articles, Witt's article presented the concepts of cross docking most thoroughly. According to his article, "cross docking can be divided into current or future cross docking. In current cross docking, material is moved directly from receiving docks to shipping docks without any intermediate staging. In future cross docking, the product will be staged someplace between receiving docks and shipping docks. Mass merchandisers are examples of current cross docking. Many of these companies move from 40 percent to 90 percent of their merchandise, especially seasonal or promotional items, via cross docking. In future cross docking, the time a product is staged can vary, but it would be hard to consider that an operation is actually cross docking if the product is staged for much more than a day." (Witt, 1998). In his article, three successful companies implementing cross docking are identified: Toyota, Supervalu, and Mitsubishi. Issues about cross docking candidates, implementation of cross docking in manufacturing and dock management are also explained in his article.

Schaffer (1998) explained the requirements for successful cross docking. He emphasized that cross docking can increase efficiency though successful implementation requires careful planning. He pointed out that most cross docking failures are due to the fact that there is an insufficient understanding of the requirements for successful cross docking and a lack of planning for the execution. In his article, the requirements for cross docking are broken down into six categories. They are: "1) partnering with other members of the distribution chain, 2) absolute confidence in the quality and availability of products, 3) communications between supply chain members, 4) communications and control within the cross docking operation, 5) personnel, equipment and facilities, and 6) operational management. In order to successfully implement cross docking, a formal program must be set up to address each of the above categories."

Drawing from published work, it is clear that cross docking systems can greatly reduce inventory, shorten the product flow time between the manufacturer and the customer, and produce better control of the distribution operation. To implement cross docking successfully, the appropriate software should be developed as well as the hardware. For example, the spotting sequences of the receiving and shipping trucks to appropriate docks are an important factor that affects the system performance. Decision on product routing is another important factor. Nevertheless, there is no reported research on these aspects of cross docking system. This lack of research motivated the study undertaken in this dissertation.

CHAPTER 3. MODEL DESCRIPTIONS

3.1 Description of the General Cross Docking Model

As presented in Section 1.2, the typical cross docking system is operated as follows:

1. Receiving trucks arrive at the receiving docks and unload products onto the receiving dock.
2. Products are scanned and verified at the receiving docks. In some cross docking systems, products are also weighed, sized and labeled at the receiving dock.
3. Products are placed on the sortation systems and sorted by destinations.
4. Products are transferred to the proper location on the shipping docks.
5. Shipping trucks load products from shipping docks and leave shipping docks.

Figure 3 shows the flow of material in a typical cross docking operation.

As presented in Section 1.5, the cross docking system of this study is operated as follows:

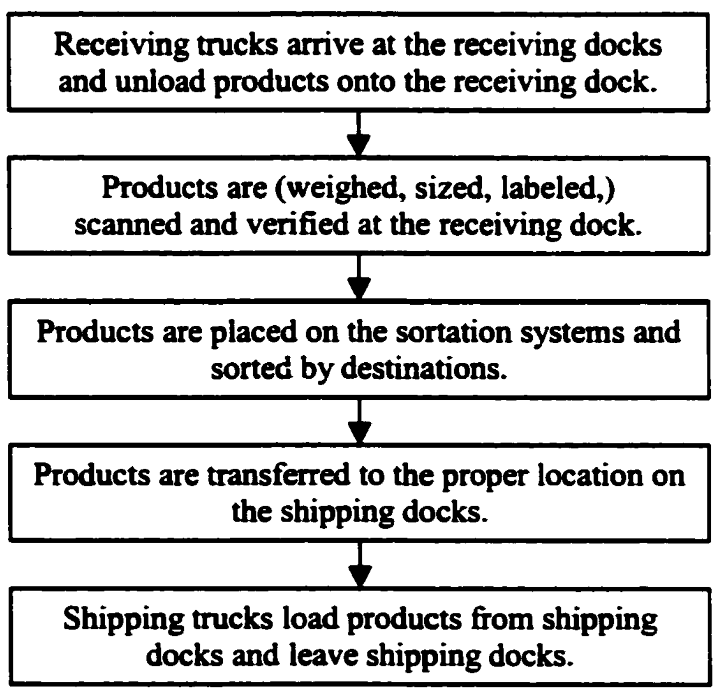


Figure 3. The Material Flow in the Cross Docking Systems

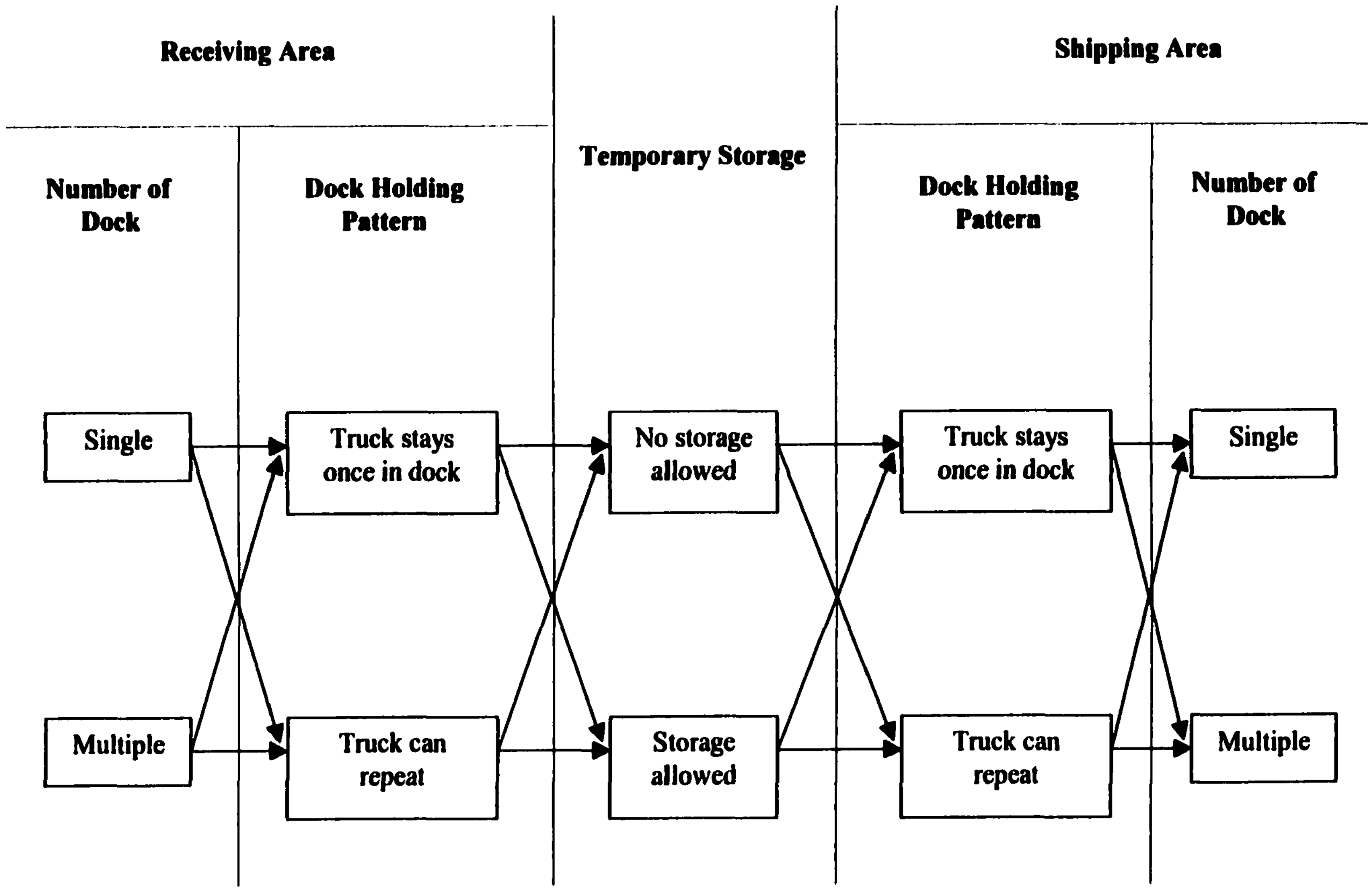
1. Receiving trucks arrive at the receiving docks and unload products onto the receiving dock.
2. Products move from the receiving dock to the shipping dock on a conveyor.
3. Shipping trucks load products from shipping docks and leave shipping docks.

The cross docking system in this research does not consider the operations inside the warehouse or distribution center such as scanning and sorting operations etc. Therefore, the arrival sequence of the products at the shipping dock is the same as their unloading sequence at the receiving dock. In other words, the order in which the products are unloaded at the receiving dock is the same as the order they arrive at the shipping dock.

3.1.1 Possible Models of a Cross Docking Operation

Depending on the facility, operating conditions or strategies employed, it is possible to generate various cross docking models. Figure 4 and Table 1 show the various models based on the number of docks available at the site, the dock holding pattern for trucks, and the existence of temporary storage. It is possible that a warehouse or distribution center may have any number of docks. A small distribution center may have one receiving dock and one shipping dock. On the other hand, a large distribution center may have hundreds of receiving docks and hundreds of shipping docks. Based on the number of docks, different operating strategies may emerge.

Similarly, for a dock holding pattern for trucks, two possible strategies can be considered. In the first strategy, whenever a truck goes into a receiving or shipping dock, it never leaves until its task is finished. In other words, all products in a receiving truck must be unloaded before the receiving truck leaves a receiving dock. Similarly, all needed products must be loaded to a shipping truck before the shipping truck leaves a shipping dock. In the second strategy, both receiving and shipping trucks can come and leave the docks repeatedly. Therefore, it is possible that a receiving truck unloads some of its products onto the receiving dock, moves out of the dock for another receiving truck, waits and goes into the receiving dock again to unload all or part of its remaining products. This operation pattern can also be similarly applied to a shipping truck.



*** A total of 32 possible models**

Figure 4. Various Models of a Cross Docking System

Table 1. Various Models of a Cross Docking System

Model Number	Receiving Area		Temporary Storage	Shipping Area		Feasibility
	Number of Dock	Dock Holding Pattern		Dock Holding Pattern	Number of Dock	
1	Single	Stay	No Storage	Stay	Single	S
2	Single	Stay	No Storage	Stay	Multiple	S
3	Single	Stay	No Storage	Repeat	Single	F
4	Single	Stay	No Storage	Repeat	Multiple	F
5	Single	Stay	Storage	Stay	Single	F
6	Single	Stay	Storage	Stay	Multiple	F
7	Single	Stay	Storage	Repeat	Single	F
8	Single	Stay	Storage	Repeat	Multiple	F
9	Single	Repeat	No Storage	Stay	Single	F
10	Single	Repeat	No Storage	Stay	Multiple	F
11	Single	Repeat	No Storage	Repeat	Single	F
12	Single	Repeat	No Storage	Repeat	Multiple	F
13	Single	Repeat	Storage	Stay	Single	F
14	Single	Repeat	Storage	Stay	Multiple	F
15	Single	Repeat	Storage	Repeat	Single	F
16	Single	Repeat	Storage	Repeat	Multiple	F
17	Multiple	Stay	No Storage	Stay	Single	S
18	Multiple	Stay	No Storage	Stay	Multiple	S
19	Multiple	Stay	No Storage	Repeat	Single	F
20	Multiple	Stay	No Storage	Repeat	Multiple	F
21	Multiple	Stay	Storage	Stay	Single	F
22	Multiple	Stay	Storage	Stay	Multiple	F
23	Multiple	Stay	Storage	Repeat	Single	F
24	Multiple	Stay	Storage	Repeat	Multiple	F
25	Multiple	Repeat	No Storage	Stay	Single	F
26	Multiple	Repeat	No Storage	Stay	Multiple	F
27	Multiple	Repeat	No Storage	Repeat	Single	F
28	Multiple	Repeat	No Storage	Repeat	Multiple	F
29	Multiple	Repeat	Storage	Stay	Single	F
30	Multiple	Repeat	Storage	Stay	Multiple	F
31	Multiple	Repeat	Storage	Repeat	Single	F
32	Multiple	Repeat	Storage	Repeat	Multiple	F

- S: Feasible in some condition, F : Feasible.
- Shaded Models: Three models studied in this dissertation.

The last consideration is the existence of temporary storage. In the first case, there is a temporary storage in front of the shipping dock. Therefore, if the appropriate shipping truck is not available when a product arrives at the shipping dock, the product can be stored in the temporary storage. In the second case, there is no temporary storage. Therefore, if the appropriate shipping truck is not available when a product arrives at the shipping dock, the product has to wait at the shipping dock until the appropriate shipping truck is available, possibly forcing the conveyor to stop.

By considering different strategies, it is possible to generate thirty-two different models as shown in Table 1. Among the thirty-two models, four models have solutions in certain cases. For the remaining twenty-eight models, different operating strategies may need to be developed.

In this research, three specific models of the cross docking systems are considered, as shown in Table 1. The detailed explanations for the three specific models are presented in Section 3.2.

3.1.2 Performance Measures

The following criteria can be used to measure the performance of the cross docking operation:

1. The number of receiving and shipping docks required.
2. Dock utilization.
3. Average time a truck spends loading and unloading.
4. Total time spent in transferring products between receiving and shipping trucks.
5. Total time required to execute a cross docking operation for a given stream of receiving and shipping trucks. This operation time is equivalent to the makespan of the system.

Depending on the performance measure adopted, different operating strategies may need to be developed. Among the above measures, the last measure is adopted in this research. Therefore, the objective of this research is to minimize the total operation time of the cross docking system or to maximize the throughput of cross docking systems.

3.1.3 Total Operation Time (Makespan)

In a sequencing or scheduling problem, total operation time is often called makespan. In this research, makespan is defined as follows: Makespan is the total operating time of the cross docking operation. The total operating time is from the moment when the first product of the first scheduled receiving truck is unloaded onto the receiving dock to the moment when the last product of the last scheduled shipping truck is loaded from the shipping dock. For the three different models studied in this research, different strategies are developed to minimize makespan for each model.

3.1.4 Influential Factors on Makespan

The factors that affect makespan of the cross docking system are as follows.

1. The layout and design of receiving and shipping docks.
2. The number of receiving and shipping docks.
3. The actual delivery and shipping schedules.
4. The mix of products and the number of products for each receiving and shipping trucks.
5. The product routing in the warehouse.
6. The material handling types used in the warehouse.
7. The fork truck task assignment if a fork truck is used as a material handling device in the warehouse.
8. The availability of trucks when they are required.
9. The delay time for truck changes. In most cross docking systems, truck changes have a significant effect on system performance.
10. The required space for temporary storage of products before they are shipped.
11. Dock holding pattern of trucks.
12. The amount of products as well as the unloading or loading sequence of product types in each truck.
13. The spotting sequences of the receiving and shipping trucks.

In this study, factors 1 to 9 are considered to be previously known information. Based on factors 10 and 11, three models will be considered separately. To minimize makespan, factors 12 and 13 will be used as decision variables. The solution of each model will seek to find the best sequences for factors 12 and 13.

3.2 Model Descriptions for Three Specific Models

In this section, the descriptions of the three specific models of interest in this research are presented. All descriptions presented in this section will be applied to all three models. If an additional description is required for a specific model, it will be presented in the Chapter in which the model is developed. All models considered in this research have only one receiving dock and one shipping dock. The three models are as follows:

Case 1 Model. There is temporary storage in front of the shipping dock. If a product arriving at the shipping dock does not need to be loaded into the current shipping truck, the product is stored in the temporary storage until the appropriate shipping truck is available. In this model, the receiving truck and the shipping truck must stay in docks once they come into docks and continue to do so (i.e., stay in the dock) until they finish their task. This corresponds to *Model 5* in Table 1.

Case 2 Model. In this model, there is no temporary storage in the warehouse or distribution center. However, both the receiving truck and the shipping truck can move in and out of the docks repeatedly during their tasks until their tasks are finished. Therefore, it is possible that a receiving truck unloads some of its products to the receiving dock, moves out, waits and goes into the receiving dock again to unload its remaining products. When the truck is out of the dock and waiting, another receiving truck could enter the dock to unload its products. This sequence can be similarly applied to the shipping truck. However, the conveyor connecting the receiving dock and the shipping dock may need to stop if a shipping truck is not available when a product arrives at the shipping dock. This is necessary because of the absence of a temporary storage. This corresponds to *Model 11* in Table 1.

Case 3 Model. In this model, there is temporary storage in front of the shipping dock and both the receiving truck and the shipping truck can move in and out during their tasks as in *Case 2 Model* until their tasks are finished. This corresponds to *Model 15* in Table 1.

3.2.1 Assumptions

The following assumptions are applied to all three models.

1. A warehouse or distribution center will be receiving and shipping at virtually the same time so that all incoming products are shipped as soon as possible.
2. All receiving and shipping trucks are available at time zero.
3. All products received must be shipped. Long term storage is not allowed.
4. The total number of receiving products for each type of products is the same as the total number of shipping products for each type of products.
5. The unloading sequence of the products from a receiving truck can be determined. For example, if a certain receiving truck carries three product types *A*, *B* and *C*. The unloading sequence of the products for the receiving truck can be *A-B-C* or *B-C-A, etc*.
6. It can be unloaded only the necessary amount of products from a receiving truck. In other words, any products loaded in a receiving truck are accessible so that only the necessary amount of products can be unloaded from a receiving truck. Suppose that a certain receiving truck has 100 units of product type *A*, 300 units of product type *B* and 150 units of product type *C*. If 50 units of product type *A* and 200 units of product type *B* are needed to be unloaded in a certain situation, only those amounts of the products can be unloaded from the receiving truck.
7. Only one unit of a product can be loaded into the shipping truck at a time. Therefore, loading products simultaneously from a receiving truck and the temporary storage into a shipping truck is prohibited.
8. The operations inside the warehouse or distribution center such as scanning and sorting operations are not considered. Therefore, the arrival sequence of the products at the shipping dock is maintained as the unloading sequence of the products at the receiving dock.
9. Delay time for truck changes is the same for all receiving and shipping trucks.
10. Moving time of products from the receiving dock to the shipping dock is the same for all products.

11. The capacity of temporary storage is unlimited. Therefore, it can be assumed that the capacity of the temporary storage can be as high as the total number of products presented in a model.
12. The following information is assumed to be previously known.
    i) Product types and the number of products loaded in a receiving truck.
    ii) Product types and the number of products needed for a shipping truck.
    iii) Loading and unloading times for the products.
    iv) Moving times of products from a receiving dock to a shipping dock.
    v) Delay time (i.e., truck change time) due to truck changes.
    vi) Delay time when a product passes through the temporary storage. For example loading time from temporary storage to a shipping truck or unloading time from a conveyor to temporary storage.

3.2.2 Expected Results

The solution of each model is expected to provide the following results.

1. The spotting sequence of the receiving trucks at the receiving dock.
2. The spotting sequence of the shipping trucks at the shipping dock.
3. The unloading sequence of the products from a receiving truck.
4. The product routings or the product assignments from receiving trucks to shipping trucks.

In other words, the solution will show how many products move from a certain receiving truck to a certain shipping truck as well as what types of products move between them. Additionally, it will also show whether products move directly from a receiving truck to a shipping truck or if they pass through the temporary storage.

In Chapter 4, the modeling and the solution approaches used for *Case 1 Model* are presented. Chapter 5 presents the modeling and the solution approaches used for *Case 2 Model*. In Chapter 6, the modeling and the solution approaches used for *Case 3 Model* are developed.

CHAPTER 4. *CASE 1* – CROSSDOCKING MODEL WITH TEMPORARY STORAGE AND DOCK NON-REPEAT TRUCK HOLDING PATTERN AT THE DOCK

4.1 Model Descriptions

In the first model of the cross docking problem studied in this research, there is temporary storage in front of the shipping dock. If a product that arrives at the shipping dock does not need to be loaded into shipping truck currently at the dock, the product can be stored in the temporary storage until the appropriate shipping truck comes into the shipping dock. In this model, both the receiving and the shipping trucks must stay in docks until they finish their task once they come into docks. Therefore, a receiving truck cannot leave the receiving dock until all of its products are unloaded onto the receiving dock. Similarly, a shipping truck cannot leave the shipping dock until all of its needed products are loaded.

The objective of this research is to find the best sequence for truck spotting for both the receiving and the shipping trucks to minimize total cross docking operation time or to maximize the throughput of the cross docking system. The product routing (i.e., the allocation of products from receiving trucks to shipping trucks) is also decided simultaneously as well as the spotting sequences of the receiving and shipping trucks.

In the *Case 1* problem, there are two sources from which products are loaded into a shipping truck. One source represents the receiving trucks and this occurs when a product transfers directly from a receiving truck to a shipping truck without passing through temporary storage. The other source is the temporary storage and this occurs when a product transfers from a receiving truck to temporary storage and is loaded from the temporary storage to the shipping truck.

The *Case 1* problem has the following characteristics.

1. Conveyor used in transferring products or items from the receiving dock to the shipping dock run continuously without stoppage. Since there is temporary storage in front of the shipping dock, arriving products at the shipping dock can be stored in temporary storage if an appropriate shipping truck is not available. Therefore, the conveyor never stops because there is no bottleneck at the shipping dock.

2. No bottleneck can occur at the receiving dock since the transfer conveyor never stops. Therefore, a receiving truck will never wait to unload its product once in dock. In other words, a receiving truck starts to unload its products as soon as it comes into a receiving dock and leaves a receiving dock as soon as all of its products are unloaded.
3. Total required unloading time of all products from all receiving trucks is independent of the receiving truck spotting sequence and the shipping truck spotting sequence. Since there is no delay in unloading products from receiving trucks, total required unloading time of all products from all receiving trucks is calculated as shown in equation below, and this is independent of the receiving and shipping truck spotting sequences:

$$\sum_{i=1}^{R}\sum_{k=1}^{N} r_{ik} u_k + (R-1)D.$$

where,

R = Number of receiving trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k that is initially loaded in receiving truck i,

u_k = Unloading time for one unit of product type k from a receiving truck,

D = Delay time for truck change.

As shown in the above equation, the total required unloading time of all products from receiving trucks is the sum of the unloading time of all products from all receiving trucks and the delay time due to receiving truck changes. However, it must be pointed out that the receiving truck spotting sequence affects makespan. The number of products passing through temporary storage, which may affect makespan, depends on both the receiving and shipping truck spotting sequences. Therefore, it can be stated that although the receiving truck sequence and shipping truck sequence do not affect the total unloading time from receiving trucks, they do, however, affect the loading time into shipping trucks and thus affect makespan.
4. In the *Case 1* problem, there are two types of delay times. The first type of delay time occurs when there is a shipping truck change. The second type of delay time occurs when the shipping truck currently in dock does not load any products from a certain receiving truck in dock or temporary storage, and waits until its needed products arrive at the

shipping dock. The change of receiving trucks at the receiving dock or the unloading products from a receiving truck to temporary storage may cause the second type of delay time. For the *Case 1* problem, the first type of delay time is the same regardless of the shipping truck spotting sequences because all shipping truck sequences have the same number of shipping truck changes which is *(S-1)*, where *S* represents the number of shipping trucks in the set. Similarly, the number of receiving truck changes are the same regardless of the receiving truck sequences. Therefore, the only factor, which can affect makespan, is the number of unloaded products from receiving trucks that transfer to the temporary storage. If the number of products sent to temporary storage decreases, the waiting time of the shipping truck at the shipping dock may decrease, thus makespan may decrease.

From the above characteristics, it can be seen that makespan will be minimized if delay time or idle time is minimized. For the *Case 1* problem, the main factor that causes idle time is the number of products passing through temporary storage. Therefore, minimizing the number of products passing through temporary storage seems to be a good strategy for minimizing makespan. However, it must be pointed out that minimizing makespan is not equivalent to minimizing the total number of products passing through temporary storage because the occurrence of idle time depends not only on the number of truck changes but also when trucks change. Depending on the receiving and shipping truck sequences, the times at which the receiving trucks or shipping trucks come into the dock or leave the dock will be changed. Moreover, there are time intervals when unloading activity from a receiving truck and loading activity into a shipping truck for the same product in a cross docking operation occur simultaneously. Therefore, in some cases, routing some of the items through temporary storage can decrease the makespan. Nevertheless, in general, minimizing the total number of products passing through temporary storage minimizes production makespan for the *Case 1* problem as presented in Section 4.3.

4.2 Model Development

To solve the cross docking problem for the *Case 1* model, five different approaches were developed. For the first approach, a mathematical model whose objective is to minimize the makespan of a cross docking operation was developed. The second approach employed complete enumeration of all possible sequences to find an optimal solution. For a small problem, the first two approaches can be used. However, it is not efficient in applying these two methods to solve medium to large problems because of the computational time required to solve the problem. Therefore, the third approach was developed to solve problems of practical sizes. The third approach employed a heuristic algorithm. The heuristic algorithm finds solutions quite fast although the solution found may not necessarily be optimal. In the fourth approach, a meta-heuristic technique was used to solve the problem. In order to test the performance of the heuristic algorithm developed in the third approach, the tabu search was applied to the *Case 1* problem and compared with the heuristic algorithm. The last approach suggested for the *Case 1* problem used the branch and bound method. It uses as the upper bound the solution found by the heuristic algorithm. This approach was also able to find the global optimal solution. For a practical size problem, the branch and bound method takes shorter time than the complete enumeration method while at the same time finds the global optimal solution.

4.2.1 Mathematical Model

For the mathematical model of the *Case 1* problem, it is assumed that unloading time from a receiving truck and loading time into a shipping truck are the same for all types of products and it takes one unit of time for one unit of product. Additionally, it is assumed that it takes one unit of time when one unit of products is unloaded from a conveyor to the temporary storage or loaded from the temporary storage into a shipping truck. It is also assumed that all operations can be carried out simultaneously except that loading operations from a conveyor into a shipping truck and from a temporary storage into a shipping truck cannot be carried out simultaneously. With the above assumptions, the following mixed integer programming model was developed for the *Case 1* problem with the objective of minimizing the makespan of a cross docking operation.

4.2.2.1 Notations

The following notations are used for the mathematical model.

Continuous Variables:

T = Makespan,

c_i = Time at which receiving truck i enters the receiving dock,

F_i = Time at which receiving truck i leaves the receiving dock,

d_j = Time at which shipping truck j enters the shipping dock,

L_j = Time at which shipping truck j leaves the shipping dock,

Integer Variables:

x_{ijk} = Number of units of product type k which transfer from receiving truck i to shipping truck j,

Binary Variables:

$$v_{ij} = \begin{cases} 1, & \text{If any products transfer from receiving truck } i \text{ to shipping truck } j \\ 0, & \text{Otherwise} \end{cases},$$

$$p_{ij} = \begin{cases} 1, & \text{If receiving truck } i \text{ preceeds receiving truck } j \text{ in the receiving truck sequence.} \\ 0, & \text{Otherwise} \end{cases},$$

$$q_{ij} = \begin{cases} 1, & \text{If shipping truck } i \text{ preceeds shipping truck } j \text{ in the shipping truck sequence} \\ 0, & \text{Otherwise} \end{cases},$$

Data:

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k that is initially loaded in receiving truck i,

s_{jk} = Number of units of product type k that is initially needed for shipping truck j,

D = Delay time for truck change,

V = Moving time of products from the receiving dock to the shipping dock,

M = Big number.

4.2.2.2 Mixed Integer Programming Model

The mixed integer programming model for the *Case 1* problem with the objective of minimizing makespan of a cross docking operation is presented below.

Mathematical Model for the *Case 1* Problem

Min T

Subject to

$$T \geq L_j, \quad \text{for all } j \tag{4–1}$$

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \quad \text{for all } i, k \tag{4–2}$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \quad \text{for all } j, k \tag{4–3}$$

$$x_{ijk} \leq M\, v_{ij}, \quad \text{for all } i, j, k \tag{4–4}$$

$$F_i \geq c_i + \sum_{k=1}^{N} r_{ik}, \quad \text{for all } i \tag{4–5}$$

$$c_j \geq F_i + D - M(1 - p_{ij}), \quad \text{for all } i, j \text{ and where } i \neq j \tag{4–6}$$

$$c_i \geq F_j + D - Mp_{ij}, \quad \text{for all } i, j \text{ and where } i \neq j \tag{4–7}$$

$$p_{ii} = 0, \quad \text{for all } i \tag{4–8}$$

$$L_j \geq d_j + \sum_{k=1}^{N} s_{jk}, \quad \text{for all } j \tag{4–9}$$

$$d_j \geq L_i + D - M(1 - q_{ij}), \quad \text{for all } i, j \text{ and where } i \neq j \tag{4–10}$$

$$d_i \geq L_j + D - Mq_{ij}, \quad \text{for all } i, j \text{ and where } i \neq j \tag{4–11}$$

$$q_{ii} = 0, \quad \text{for all } i \tag{4–12}$$

$$L_j \geq c_i + V + \sum_{k=1}^{N} x_{ijk} - M(1 - v_{ij}), \quad \text{for all } i, j \tag{4–13}$$

$$\text{all variables} \geq 0.$$

Constraint (4–1) makes makespan equal the time the last scheduled shipping truck leaves the shipping dock. *Constraint (4–2)* ensures that the total number of units of product type k that transfer from receiving truck i to all shipping trucks is exactly the same as the number of units of product type k that was initially loaded in receiving truck i. Similarly, *constraint (4–3)* ensures that the total number of units of product type k that transfer from all receiving trucks to shipping truck j is exactly the same as the number of units of product type k needed for shipping truck j. *Constraint (4–4)* just enforces the correct relationship between the x_{ijk} variables and the v_{ij} variables.

Constraints (4–5) to *(4–7)* make a valid sequence for arriving and departing times for the receiving trucks based on their order. *Constraint (4–8)* ensures that no receiving truck can precede itself in the receiving truck sequence. Similar to *constraints (4–5)* to *(4–7)* for receiving trucks, *constraints (4–9)* to *(4–11)* function in a similar manner for the shipping trucks. Similar to *constraint (4–8)*, *constraint (4–12)* ensures that no shipping truck can precede itself in the shipping truck sequence. *Constraint (4–13)* connects the leaving time for a shipping truck to the arriving time of a receiving truck if any products or items are transferred between the trucks.

The number of decision variables for this mixed integer programming model is $RS(N+3)+2(R+S)+1$. The decision variables consist of $3RS$ of binary variables, RSN of integer variables and $2(R+S)+1$ of continuous variables. The number of constraints is $2(R^2+S^2)+R(S+N)+S(RN+N+1)$, including $2(R^2+S^2)+R(S-1)$ of inequality constraints and $(R+S)(N+1)+RSN$ of equality constraints.

4.2.2 Complete Enumeration Method

In the *Case 1* problem, a receiving truck stays in the receiving dock until it finishes its unloading operation. Therefore, each receiving truck appears only once in the receiving truck sequence. Similarly, each shipping truck appears only once in the shipping truck sequence because a shipping truck stays in the shipping dock until it loads all its needed products. Therefore, if all possible combinations of the receiving and shipping truck sequences are enumerated, the optimal solutions can be found. The total number of possible sequences is $(R!)(S!)$ for the complete or exhaustive enumeration method of the *Case 1* problem.

For a small problem, it is possible to find an optimal solution with this method. For example, suppose that the number of receiving trucks is four and the number of shipping trucks is five. Then, the total number of possible sequences for this problem will be $(4!)(5!)$ = 2880. Therefore, if all the 2880 sequences are examined, the optimal solution to the problem will be found. However, this method is not practical for medium to large size problems. For example, suppose that the number of receiving trucks and shipping trucks are each ten, respectively. Then, the total number of possible sequences will be $(10!)(10!) = 1.3\times10^{13}$. In this case, it is not practical to solve this problem by enumerating all possible sequences. Therefore, what is required is a method that finds solutions within reasonable amount of time. The next section describes the heuristic method developed to find solutions within reasonable amount of time.

The reason why the complete enumeration approach is implemented in this research is to provide a basis to benchmark the performance of the heuristic algorithm. For small size problems, this method is able to find the worst solution and the average solution of all possible sequences as well as the optimal solution because it enumerates all possible sequences. Solutions obtained from the heuristic algorithm can then be compared with the solutions obtained by the enumeration technique to test the algorithmic performance.

4.2.3 Heuristic Method

For the heuristic algorithms of the *Case 1* problem, the same assumptions were made as in the mathematical model presented in section 4.2.1. Therefore, it is assumed that all unloading times and loading times for all types of products are the same and it takes one unit of time. The case in which the above assumptions are relaxed is presented in Section 4.2.3.5.

Figure 5 shows the flow of products in a cross docking operation for the *Case 1* problem. There are two routes for transferring products from a receiving truck to a shipping truck in the *Case 1* problem.

1. A product transfers from a receiving truck to a shipping truck directly without passing through temporary storage.
2. A product transfers from a receiving truck to temporary storage first and is loaded from temporary storage to a shipping truck.

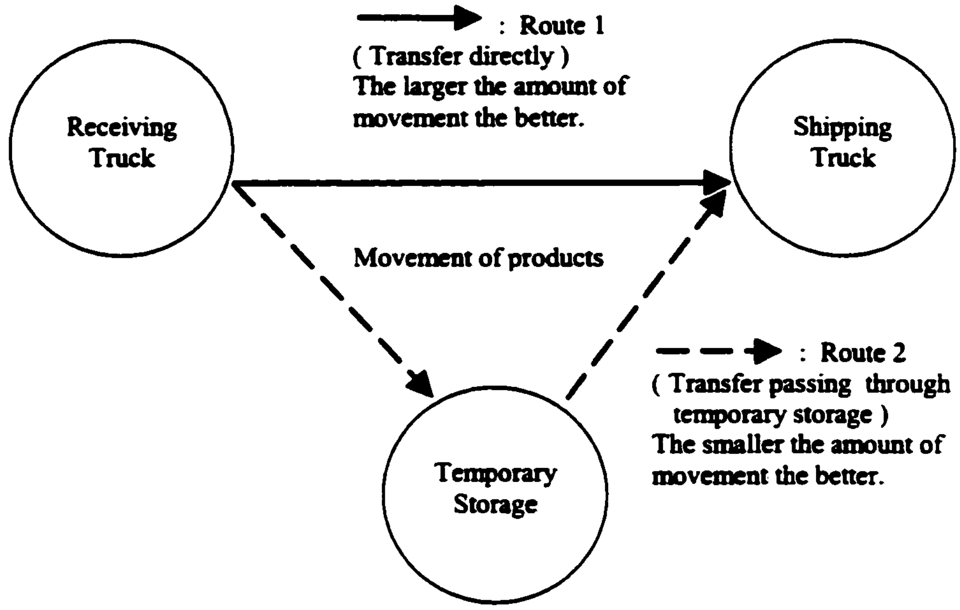


Figure 5. Two Routes for Transferring Products from a Receiving Truck to a Shipping Truck

In order to minimize makespan for the *Case 1* problem, it is a good strategy to transfer as many products as possible directly from a receiving truck to a shipping truck without passing through the temporary storage. Conversely, it is a good strategy to transfer the fewest number of products passing through the temporary storage. The main idea of the heuristic algorithms developed in this research came from the above premises.

The heuristic algorithm developed for the *Case 1* problem consists of two major stages of decision. In the first stage, the best associate receiving trucks are found for each unscheduled shipping truck based on the associate receiving truck selection strategy. The associate receiving trucks for a shipping truck are defined as the sets of receiving trucks that carry enough products to satisfy the requirements of the shipping truck. Therefore, many different types of associate receiving trucks can exist for the same shipping truck.

In the second stage, one of the unscheduled shipping trucks is selected based on the shipping truck selection strategy. The selected shipping truck is placed in the next available shipping truck sequence. The associate receiving trucks for the selected shipping truck are placed in the next available receiving truck sequence. Once a shipping truck and its associate receiving trucks are scheduled, the lists of unscheduled receiving trucks and unscheduled shipping trucks are updated. Next, for each unscheduled shipping truck in the updated list, a new set of its associate receiving truck is formed from the updated unscheduled receiving truck list. Again, based on the selection strategy employed, a shipping truck and its associate receiving trucks are selected and scheduled. Once a shipping truck and its associates are selected and scheduled, the lists of unscheduled shipping and receiving trucks are again updated. The process of selection, scheduling and updating is continued until all trucks are scheduled.

4.2.3.1 Notations

For the heuristic algorithm of the *Case 1* problem, the following notations are used:

Data:

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k which is initially loaded in receiving truck i,

s_{jk} = Number of units of product type k which is initially needed for shipping truck j,

D = Delay time for truck change,

V = Moving time of products from the receiving dock to the shipping dock,

Truck:

t^r_i = Receiving truck i,

t^s_j = Shipping truck j,

t^r_L = Last scheduled receiving truck in set T^r,

Set:

T^r = Ordered set of scheduled receiving trucks,

T^s = Ordered set of scheduled shipping trucks,

U^r = Set of unscheduled receiving trucks,

U^s = Set of unscheduled shipping trucks,

A^R_j = Ordered set of associate receiving trucks for shipping truck j,

Number of Products:

r'_{ik} = Number of units of product type k which are loaded in receiving truck i in a given iteration of the algorithm,

s'_{jk} = Number of units of product type k which are needed for shipping by truck j in a given iteration of the algorithm,

r_{Lk} = Number of units of product type k which remains in the last scheduled receiving truck after sending products to the last scheduled shipping truck,

t_k = Number of product type k which is stored in temporary storage in a given iteration of the algorithm,

$p^{RS}{}_{ij}$ = Total number of products which transfers directly from receiving truck i to shipping truck j when they are scheduled,

$p^{RT}{}_{ij}$ = Total number of products which transfers from receiving truck i to temporary storage when receiving truck i and shipping truck j are scheduled,

$p^{AS}{}_{j}$ = Total number of products which transfers from the associate receiving trucks of shipping truck j to shipping truck j $\left(i.e.\ p^{AS}{}_{j} = \sum_{k=1}^{N} s_{jk} \right)$,

$p^{AT}{}_{j}$ = Total number of products which transfers from the associate receiving trucks of the shipping truck j to temporary storage,

Ratio:

$p^{T/S}{}_{ij}$ = Ratio of the number of products transferring from receiving truck i into temporary storage to the number of products transferring directly from receiving truck i into shipping truck j,

$p^{A(T/S)}{}_{j}$ = Ratio of the number of products transferring from associate receiving trucks $A^{s}{}_{j}$ of shipping truck j into temporary storage to the number of products needed for shipping truck j,

Time:

$p^{TM}{}_{j}$ = The amount of time shipping truck j stays at the shipping dock.

4.2.3.2 Selection Strategies for the Best Associate Receiving Trucks

As mentioned earlier, the heuristic algorithm developed in this research consists of two major stages. The first stage of the heuristic algorithm is to find the best associate receiving trucks for each unscheduled shipping truck. To find the best associate receiving trucks for a certain shipping truck, the algorithm follows the steps presented below: first, each unscheduled receiving truck is matched with the shipping truck. For each pair of an unscheduled receiving truck and the shipping truck, the number of products that transfer directly from the receiving truck to the shipping truck is calculated. The number of products that transfer from the receiving truck to temporary storage is also calculated. Thereafter, one of the unscheduled receiving trucks is chosen based on one of the following strategies:

I. *Associate Receiving Truck Selection Strategy 1*

- The receiving truck that transfers the smallest number of products to temporary storage is chosen.

II. *Associate Receiving Truck Selection Strategy 2*

- The receiving truck that transfers the largest number of products to the shipping truck is chosen.

III. *Associate Receiving Truck Selection Strategy 3*

- The receiving truck that has the smallest ratio of the number of products transferring from a receiving truck into temporary storage to the number of products transferring from a receiving truck into the shipping truck is chosen.

After one of the unscheduled receiving trucks is selected, the needed products for the shipping truck are updated. Then, the procedure is continued until the shipping truck loads all of its needed products. The detailed explanations about the three selection strategies to find the best associate receiving trucks A^R_j for shipping truck t^s_j are presented below.

I. *Associate Receiving Truck Selection Strategy 1*

For each $t^r_i \in U^r$ and $t^r_i \notin A^R_j$, the number of products transferring from receiving truck t^r_i to temporary storage is calculated; (i.e. p^{RT}_{ij} is calculated). Calculation of p^{RT}_{ij} is divided into two different cases:

i) Shipping truck t^s_j loads all of its needed products after it receives products from receiving truck t^r_i. In this case, it is easy to think that no products need to transfer to temporary storage because only the needed products for shipping truck t^s_j can be unloaded from receiving truck t^r_i. However, this is not always true since after the current shipping truck t^s_j leaves, receiving truck t^r_i may still have some products left to be unloaded. Depending upon the next scheduled shipping truck, some products in receiving truck t^r_i may transfer to temporary storage. Therefore, it is needed to calculate how many units of products transfer to temporary storage for each possible case of the next scheduled shipping truck $t^s_{j'}$. This situation can be expressed as follows:

If $\sum_{k=1}^{N}\left[\max\{s'_{jk} - r'_{ik}, 0\}\right]$ is zero, then calculate p^{RT}_{ij} as follows:

$$p^{RT}_{ij} = \min_{\substack{1 \le j' \le S \\ j' \ne j \\ t^s_{j'} \in U^S}} \left[\sum_{k=1}^{N} \max\{(r'_{ik} - s'_{jk}) - s_{j'k}, 0\}\right]. \tag{4-14}$$

In *equation (4-14)*, the term, $(r'_{ik}-s'_{jk})$, represents the remaining products in receiving truck t^r_i after sending its products to shipping truck t^s_j. Therefore, the term $\sum_{k=1}^{N} \max\{(r'_{ik} - s'_{jk}) - s_{j'k}, 0\}$ will be zero if all remaining products in receiving truck t^r_i transfer to the next scheduled shipping truck $t^s_{j'}$. However, it will have a positive value if the next scheduled shipping truck $t^s_{j'}$ does not need some products from receiving truck t^r_i, thus the products need to transfer to temporary storage. Therefore, *equation (4-14)* represents the least number of products that transfer from receiving truck t^r_i to temporary storage where shipping truck $t^s_{j'}$ is the next scheduled shipping truck.

ii) Shipping truck t^s_j needs to load more products after it receives products from receiving truck t^r_i. In this case, all remaining products in receiving truck t^r_i must transfer to temporary storage. In other words, if $\sum_{k=1}^{N}\left[\max\{s'_{jk} - r'_{ik}, 0\}\right]$ is positive, then p^{RT}_{ij} is calculated as follows:

$$p^{RT}_{ij} = \sum_{k=1}^{N}\left[\max\{r'_{ik} - s'_{jk}, 0\}\right]. \tag{4-15}$$

After calculating all p^{RT}_{ij} for each unscheduled receiving truck i, the receiving truck $t^r_{i^*}$ that has the smallest p^{RT}_{ij} is chosen. If there is a tie, receiving truck $t^r_{i^*}$ that has the largest p^{RS}_{ij} is chosen. In *Strategy 2* for the associate receiving truck selection strategy, p^{RS}_{ij} is defined (in *equation (4-16)*). Then, the selected receiving truck $t^r_{i^*}$ is placed at the end of the sequence in set A^R_j.

$$A^R_j = \{ \ldots, t^r_{i^*}\}.$$

Then, s'_{jk} is updated. The above procedure is continued until shipping truck t^s_j loads all of its needed products (i.e. the above procedure is continued until $\sum_{k=1}^{N} s'_{jk} = 0$).

II. *Associate Receiving Truck Selection Strategy 2*

For each $t^r_i \in U^r$ and $t^r_i \notin A^R_j$, the number of products transferring from receiving truck t^r_i to shipping truck t^s_j is calculated as follows; (i.e. p^{RS}_{ij} is calculated as follows):

$$p^{RS}_{ij} = \sum_{k=1}^{N} \left[\min\left\{r'_{ik}, s'_{jk}\right\}\right]. \tag{4-16}$$

After calculating all p^{RS}_{ij} for each unscheduled receiving truck i, the receiving truck $t^r_{i^*}$ that has the largest p^{RS}_{ij} is chosen. If there is a tie, the receiving truck $t^r_{i^*}$ that has the smallest p^{RT}_{ij} is chosen. The selected receiving truck $t^r_{i^*}$ is placed at the end of the sequence in set A^R_j.

$$A^R_j = \{ \ldots, t^r_{i^*}\}.$$

Then, s'_{jk} is updated and the above procedure is continued until $\sum_{k=1}^{N} s'_{jk} = 0$.

III. *Associate Receiving Truck Selection Strategy 3*

For each $t^r_i \in U^r$ and $t^r_i \notin A^R_j$, the ratio of the number of products transferring from receiving truck t^r_i into temporary storage to the number of products transferring from receiving truck t^r_i into shipping truck t^s_j is calculated as follows; (i.e. $p^{T/S}_{ij}$ is calculated as follows):

$$p^{T/S}_{ij} = \frac{p^{RT}_{ij}}{p^{RS}_{ij}}. \tag{4-17}$$

The value for p^{RT}_{ij} can be calculated from *equations (4-14)* or *(4-15)* and the value for p^{RS}_{ij} can be calculated from *equations (4-16)*.

After calculating all $p^{T/S}_{ij}$ for each unscheduled receiving truck i, the receiving truck $t^r_{i^*}$ that has the smallest $p^{T/S}_{ij}$ is chosen. If there is a tie, the receiving truck $t^r_{i^*}$ that has the smallest p^{RT}_{ij} is chosen. The selected receiving truck $t^r_{i^*}$ is placed at the end of the sequence in set A^R_j.

$$A^R_j = \{ \ldots, t^r_{i^*} \}.$$

Then, s'_{jk} is updated and the above procedure is continued until $\sum_{k=1}^{N} s'_{jk} = 0$.

Depending upon the associate receiving truck selection strategy, different associate receiving trucks can be formed for the same shipping truck. Because the associate receiving trucks of the next scheduled shipping truck are scheduled as the next scheduled receiving trucks, choosing a different associate receiving truck selection strategy may affect the receiving truck sequence. Meanwhile, as can be seen later, a shipping truck is selected as the next scheduled shipping truck based on one of shipping truck selection strategies. The shipping truck selection strategies use the information about the amount of flow or time associated with the associate receiving trucks and the shipping truck. Therefore, choosing a different associate receiving truck selection strategy may affect the shipping truck sequence as well. Because choosing a different associate receiving truck selection strategy may affect both the receiving and shipping truck sequences, it can also affect makespan.

To illustrate the effect of the associate receiving truck selection strategy, consider Example 1 as described below. Example 1 has four receiving trucks, three shipping trucks and four product types. Information about each receiving truck and shipping truck is presented in Table 2. It is assumed that all loading and unloading times for all types of products are the same and are one unit of time in duration. Then, the associate receiving trucks for shipping truck t^s_1 can be found as follows. The first step of the algorithm is to calculate P^{RT}_{i1}, P^{RS}_{i1} and $P^{T/S}_{i1}$, where i = 1, 2, 3 or 4, based on the associate receiving truck selection strategy for each unscheduled receiving truck t^r_1, t^r_2, t^r_3 and t^r_4. Table 3 shows the selected receiving truck in the first iteration based on the associate receiving truck selection strategy. As presented in Table 3, various receiving trucks are selected depending on the associate receiving truck selection strategy used. For example, receiving truck t^r_1 is selected for *Strategy 1*. If *Strategy 2* is used, receiving truck t^r_2 will be selected. Receiving truck t^r_3 is selected for *Strategy 3*. After one of the receiving trucks is selected in the first iteration, the needed products for shipping truck t^s_1 is updated and the procedure is continued until all needed products for shipping truck t^s_1 are loaded. The complete solution procedure for finding the best associate receiving trucks is presented in Appendix A.

Table 2. Example Set 1 to Illustrate Associate Receiving Truck Selection Strategy

Receiving Truck

Truck	Product	Quantity
1	1	100
	3	50
2	1	100
	2	50
	3	100
3	1	100
	2	40
	4	60
4	2	100
	4	200

Shipping Truck

Truck	Product	Quantity
1	1	100
	2	100
2	1	120
	3	110
	4	160
3	1	80
	2	90
	3	40
	4	100

Table 3. Selected Associate Receiving Truck for Shipping Truck 1 in the First Iteration based on the Associate Receiving Truck Selection Strategy

Trucks	Strategy 1	Strategy 2	Strategy 3
t^r_1 & t^s_1	$P^{RT}_{11} = 50^*$ (50 units of Product 3)	$P^{RS}_{11} = 100$ (100 units of Product 1)	$P^{T/S}_{11} = 0.5$ (= 50/100)
t^r_2 & t^s_1	$P^{RT}_{21} = 100$ (100 units of Product 3)	$P^{RS}_{21} = 150^*$ (100 units of Product 1 & 50 units of Product 2)	$P^{T/S}_{21} = 0.67$ (= 100/150)
t^r_3 & t^s_1	$P^{RT}_{31} = 60$ (60 units of Product 4)	$P^{RS}_{31} = 140$ (100 units of Product 1 & 40 units of Product 2)	$P^{T/S}_{31} = 0.43^*$ (= 60/140)
t^r_4 & t^s_1	$P^{RT}_{41} = 200$ (200 units of Product 4)	$P^{RS}_{41} = 100$ (100 units of Product 2)	$P^{T/S}_{41} = 2.00$ (= 200/100)
Selected Receiving Truck	t^r_1	t^r_2	t^r_3

4.2.3.3 Selection Strategies for the Next Scheduled Shipping Truck

The second stage of the heuristic algorithm is to select the best shipping truck and its associate receiving trucks. The selected shipping truck is placed at the end of the shipping truck sequence while the associate receiving trucks of the selected shipping truck are placed at the end of the receiving truck sequence. For each unscheduled shipping truck and its associate receiving trucks, the number of products that transfer from the associate receiving trucks to temporary storage is calculated. The amount of time the shipping truck stays at the shipping dock and the number of products that are initially needed for the shipping truck are also calculated based on the shipping truck selection strategy. Then, the shipping truck and its associate receiving trucks are selected based on one of the following strategies:

I. *Shipping Truck Selection Strategy 1*

- The shipping truck and its associate receiving trucks that transfer the smallest number of products from the associate receiving trucks to temporary storage are chosen.

II. *Shipping Truck Selection Strategy 2*

- The shipping truck that stays the shortest time in the shipping dock is chosen. The associate receiving trucks of the selected shipping truck are chosen.

III. *Shipping Truck Selection Strategy 3*

- The shipping truck and its associate receiving trucks that have the smallest ratio of the number of products transferring from the associate receiving trucks into temporary storage to the number of products needed for the shipping truck are chosen.

Suppose that the associate receiving trucks in set A^R_j for the shipping truck t^s_j are formed as follows:

$$A^R_j = \{t^r_{[1]}, t^r_{[2]}, \ldots, t^r_{[z]}\}.$$

Note that $t^r_{[1]} = t^r_L$ if $T^r \neq \varnothing$ and $\sum_{k=1}^{N} r_{Lk} \neq 0$. In other words, if there is any scheduled receiving truck in the receiving truck sequence and the last scheduled receiving truck still unloads its remaining products after sending products to the last scheduled shipping truck, the last scheduled receiving truck automatically becomes the first associate receiving truck for the next scheduled shipping truck. In this case, $t^r_{[1]} \in T^r$ and $t^r_{[2]}, t^r_{[3]}, \ldots, t^r_{[z]} \in U^r$. Otherwise, $t^r_{[1]}, , t^r_{[2]}, \ldots, t^r_{[z]} \in U^r$.

I. *Shipping Truck Selection Strategy 1*

For each $t^s_j \in U^s$ and its associate receiving trucks in set A^R_j, the total number of products which transfer from the associate receiving trucks in set A^R_j to temporary storage is calculated as follows; (i.e. p^{AT}_j is calculated as follows):

$$p^{AT}{}_j = \sum_{i=1}^{z} p^{RT}{}_{[i]j} \, . \tag{4-18}$$

The term $p^{RT}_{[i]j}$ presents the number of products transferring from receiving truck $t^r_{[i]}$ in set A^R_j to temporary storage. The term $p^{RT}_{[i]j}$ is calculated from *equation (4-14)* or *(4-15)* in Section 4.2.3.2.

After calculating p^{AT}_j for all $t^s_j \in U^s$, the shipping truck $t^s_{j^*}$ that has the smallest p^{AT}_j is chosen. If there is a tie, the shipping truck $t^s_{j^*}$ that has the largest p^{AS}_j is chosen. The term p^{AS}_j means that total number of products which transfer from the associate receiving trucks of the shipping truck t^s_j to the shipping truck t^s_j $\left(i.e.\ p^{AS}{}_j = \sum_{k=1}^{N} s_{jk} \right)$. The selected shipping truck $t^s_{j^*}$ is placed at the end of the sequence in set T^s.

$$T^s = \{ \ldots, t^s_{j^*} \}.$$

Then, the associate receiving trucks of $t^s_{j^*}$ are identified and placed at the end of the sequence in set T^r; (i.e. set $A^R_{j^*}$ is identified and placed at the end of the sequence in set T^r).

$$T^r = \{ \ldots, A^R_{j^*} \}.$$

The order of receiving trucks in set $A^R_{j^*}$ must be maintained when they are placed in set T^r.

II. *Shipping Truck Selection Strategy 2*

For each $t^s_j \in U^s$ and its associate receiving trucks in set A^R_j, the amount of time shipping truck t^s_j stays at the shipping dock is calculated; (i.e. p^{TM}_j is calculated). The staying time of shipping truck t^s_j is defined as the amount of time spent from the moment it enters the shipping dock to the moment it leaves the shipping dock. Because of the dynamic characteristics of a cross docking operation, however, it is difficult to find the exact staying time of shipping truck t^s_j at the dock unless all receiving and shipping truck schedules are made. Therefore, the approximation for the staying time of shipping truck t^s_j as presented in *equation (4-19)* is used to find the staying time of shipping truck t^s_j.

$$p^{TM}{}_j = p^{AS}{}_j + p^{AT}{}_j + (z-1)D, \tag{4-19}$$

The first term, $p^{AS}{}_j$, presents the total time required for loading products to shipping truck t^s_j. The second term, $p^{AT}{}_j$, presents the total unloading time of products that transfer from the associate receiving trucks to temporary storage and is calculated from *equation (4-18)*. This term, $p^{AT}{}_j$, represents the idle time of shipping truck t^s_j because the maximum idle time which can be occurred for shipping truck t^s_j is the value of $p^{AT}{}_j$. The last term, *(z-1)D*, presents that delay time for receiving truck changes. Because the number of receiving trucks of the associate receiving trucks in set $A^R{}_j$ is *(z)*, the total number of receiving truck changes is *(z−1)*. If there are more than one associate receiving truck for shipping truck t^s_j, the delay time for the receiving truck changes needs to be considered. Therefore, the last term of *equation (4-19)* will be positive. Otherwise, the last term of *equation (4-19)* is zero because there is no receiving truck change. Note that $p^{AS}{}_j = \sum_{k=1}^{N} s_{jk}$.

After calculating $p^{TM}{}_j$ for all $t^s_j \in U^s$, the shipping truck $t^s_{j^*}$ that has the smallest $p^{TM}{}_j$ is chosen. If there is a tie, the shipping truck $t^s_{j^*}$ that has the smallest $p^{AT}{}_j$ is chosen. Then, the selected shipping truck $t^s_{j^*}$ is placed at the end of the sequence in set T^s.

$$T^s = \{\ \ldots, t^s_{j^*}\}.$$

The next steps are the same as in *Shipping Truck Selection Strategy 1*. Set $A^R{}_{j^*}$ is identified and placed at the end of the sequence in set T^r.

$$T^r = \{\ \ldots, A^R{}_{j^*}\}.$$

The order of receiving trucks in set $A^R{}_{j^*}$ must be maintained when they are placed in set T^r.

III. *Shipping Truck Selection Strategy 3*

For each $t^s_j \in U^s$ and its associate receiving trucks in set $A^R{}_j$, the ratio of the number of products transferring from the associate receiving trucks in set $A^R{}_j$ into temporary storage to the number of products needed for the shipping truck t^s_j is calculated as follows; (i.e. $p^{A(T/S)}{}_j$ is calculated as follows):

$$p^{A(T/S)}{}_j = \frac{p^{AT}{}_j}{p^{AS}{}_j}. \tag{4-20}$$

The value for $p^{AT}{}_j$ is obtained according to *equation (4-18)* and $p^{AS}{}_j = \sum_{k=1}^{N} s_{jk}$.

After calculating $p^{A(T/S)}{}_j$ for all $t^s_j \in U^s$, the shipping truck $t^s_{j^*}$ that has the smallest $p^{A(T/S)}{}_j$ is chosen. If there is a tie, the shipping truck $t^s_{j^*}$ that has the smallest $p^{AT}{}_j$ is chosen. The selected shipping truck $t^s_{j^*}$ is placed at the end of the sequence in set T^s.

$$T^s = \{\ \ldots, t^s_{j^*}\}.$$

The next steps are the same as in *Shipping Truck Selection Strategy 1*. Set $A^R{}_{j^*}$ is identified and placed at the end of the sequence in set T^r.

$$T^r = \{\ \ldots, A^R{}_{j^*}\}.$$

The order of receiving trucks in set $A^R{}_{j^*}$ must be maintained when they are placed in set T^r.

Depending upon the shipping truck selection strategy, a different shipping truck can be chosen as the next scheduled shipping truck. Scheduling of the shipping trucks affects the scheduling of the receiving trucks because each shipping truck may have different associate receiving trucks. Therefore, implementing different shipping truck selection strategies can affect makespan.

4.2.3.4 Heuristic Algorithm

In this section, the complete heuristic algorithms are presented. They are based on the concepts presented in the previous sections. The general concepts of the algorithms are as described below.

The heuristic algorithm consists of two major stages of decision. In the first stage, the best associate receiving trucks are found for each unscheduled shipping truck. In the second stage, one of the unscheduled shipping trucks and its associate receiving trucks are selected and scheduled. Once a shipping truck and its associate receiving trucks are scheduled, the lists of unscheduled receiving trucks and unscheduled shipping trucks are updated. Next, for each unscheduled shipping truck in the updated list, a new set of its associate receiving trucks is formed from the updated unscheduled receiving truck list. Again, based on the strategy employed, a shipping truck and its associate receiving trucks are selected and scheduled. Once a shipping truck and its associates are selected and scheduled, the list of unscheduled shipping and receiving trucks are again updated. The process of selection, scheduling and updating is continued until all trucks are scheduled.

The detailed algorithmic steps of the heuristic algorithm for the *Case 1* problem are as shown below.

HEURISTIC ALGORITHM FOR THE *CASE 1* PROBLEM

STEP 1

Set $T^r = \varnothing$, $T^s = \varnothing$, $U^r = \{t^r_1, t^r_2, t^r_3, \ldots, t^r_R\}$ and $U^s = \{t^s_1, t^s_2, t^s_3, \ldots, t^s_S\}$. Set $t_k = 0$, for $k = 1, 2, \ldots, N$.

STEP 2

For each shipping truck $t^s_j \in U^s$, find the best associate receiving trucks as follows:

2a

$$A^R_j = \varnothing, p^{AT}_j = 0, p^{AS}_j = \sum_{k=1}^{N} s_{jk}\ .$$

2b

If there is no scheduled receiving truck in the receiving truck sequence or no products remain in the last scheduled receiving truck after sending products to the last scheduled shipping truck, then set $s'_{jk} \leftarrow s_{jk}$ for $k = 1, 2, \ldots, N$; (i.e. if $T^r = \varnothing$ or $\sum_{k=1}^{N} r_{Lk} = 0$, $s'_{jk} \leftarrow s_{jk}$). Go to *2d* in *Step 2*.

Otherwise, do the following:

$$s'_{jk} \leftarrow \max\{s_{jk} - r_{Lk}, 0\}, \text{ for } k = 1, 2, \ldots, N.$$

$$A^R_j = \{t^r_L\}.$$

<u>2c</u>

If t^s_j needs to load more products, calculate $p^{AT}{}_j$. In other words, if $\sum_{k=1}^{N} s'_{jk} \neq 0$, calculate $p^{AT}{}_j$ as follows:

$$p^{AT}{}_j = \sum_{k=1}^{N} \max\{r_{Lk} - s_{jk}, 0\}$$

Go to *2d* in *Step 2*.

Otherwise (i.e. if $\sum_{k=1}^{N} s'_{jk} = 0$), shipping truck t^s_j loads all of its needed products from the last scheduled receiving truck. Go to *2h* in *Step 2*.

<u>2d</u>

If shipping truck t^s_j can loads all of its needed products from temporary storage, the associate receiving trucks for the shipping truck t^s_j are found. In other words, if $\sum_{k=1}^{N} [\max\{s'_{jk} - t_k, 0\}] = 0$, go to *2h* in *Step 2*. Otherwise, go to *2e* in *Step 2*.

<u>2e</u>

For each receiving truck $t^r_i \in U^r$ and $t^r_i \notin A^R{}_j$, do one of the following calculations based on the associate receiving truck selection strategy employed:

i) Associate Receiving Truck Selection Strategy 1

Calculate the number of products transferring from receiving truck t^r_i to temporary storage. In other words, calculate $p^{RT}{}_{ij}$ as presented in *Equation (4-14)* or *(4-15)*.

ii) Associate Receiving Truck Selection Strategy 2

Calculate the number of products transferring from receiving truck t^r_i to shipping truck t^s_j. In other words, calculate $p^{RS}{}_{ij}$ as presented in *Equation (4-16)*.

iii) Associate Receiving Truck Selection Strategy 3

Calculate the ratio of the number of products transferring from receiving truck t^r_i into temporary storage to the number of products transferring from receiving truck t^r_i into shipping truck t^s_j. In other words, calculate $p^{T/S}{}_{ij}$ as presented in *Equation (4-17)*.

2f

Choose the receiving truck $t^r_{i^*} \in U^R$ based on one of the associate receiving truck selection strategies employed as presented in Section 4.2.3.2. Place the selected receiving truck $t^r_{i^*}$ at the end of the sequence in set A^R_j.

$$A^R_j = \{\ \ldots, t^r_{i^*}\}.$$

2g

Update s'_{jk} based on the selected receiving truck as follows.

$$s'_{jk} \leftarrow \max\{s'_{jk} - r_{i^*k}, 0\},\ \text{ for } k = 1, 2, \ldots, N.$$

If t^s_j needs to load more products, go to *2d* in *Step 2* to find the next associate receiving truck; (i.e. $\sum_{k=1}^{N} s'_{jk} \neq 0$, go to *2d* in *Step 2*). Otherwise, go to *2h* in *Step 2*.

2h

Check whether there is any shipping truck that does not have its associate receiving trucks. If there is any shipping truck that does not have its associate receiving trucks, go to the beginning of *Step 2* to identify its associate receiving trucks. Otherwise, go to *Step 3*.

STEP 3

For each shipping truck $t^s_j \in U^S$ and its associate receiving trucks A^R_j obtained from *Step 2*, do one of the following calculations based on the shipping truck selection strategy employed:

3a Shipping Truck Selection Strategy 1

Calculate total number of products that transfer from the associate receiving trucks in set A^R_j to temporary storage. In other words, calculate p^{AT}_j as presented in *Equation (4-18).*

3b Shipping Truck Selection Strategy 2

Calculate the amount of time the shipping truck t^s_j stays at the shipping dock. In other words, calculate p^{TM}_j as presented in *Equation (4-19).*

3c Shipping Truck Selection Strategy 3

Calculate the ratio of the number of products transferring from the associate receiving trucks in set A^R_j into temporary storage to the number of products needed for the shipping truck t^s_j. In other words, calculate $p^{A(T/S)}_j$ as presented in *Equation (4-20)*.

STEP 4

Choose the shipping truck $t^s_{j^*} \in U^s$ based on one of the shipping truck selection strategies employed as presented in Section 4.2.3.3. Remove shipping truck $t^s_{j^*}$ from set U^s and place it at the end of the sequence in set T^s.

$$T^s = \{ \ldots, t^s_{j^*}\}.$$

STEP 5

Identify the associate receiving trucks of $t^s_{j^*}$ that are found in *Step 2*. In other words, identify $A^R_{j^*}$. Remove the receiving trucks in set $A^R_{j^*}$ from set U^r and place them at the end of the sequence in set T^r.

$$T^r = \{ \ldots, A^R_{j^*}\}.$$

The order of receiving trucks in set $A^R_{j^*}$ must be maintained when they are placed in set T^r.

STEP 6

Update the values of t^r_L, r_{Lk}, and t_k based on the selected shipping truck and its associate receiving trucks.

STEP 7

If $U^r = U^s = \varnothing$, stop; the receiving and shipping truck sequences are found. Set T^r presents the receiving truck sequence while set T^s shows the shipping truck sequence. Otherwise, go to *Step 2*.

Figures 6 and 7 describe the algorithmic steps of the heuristic algorithm for the *Case 1* problem. The complete solution procedure of Example 1 in Section 4.2.3.2 is presented in Appendix A.

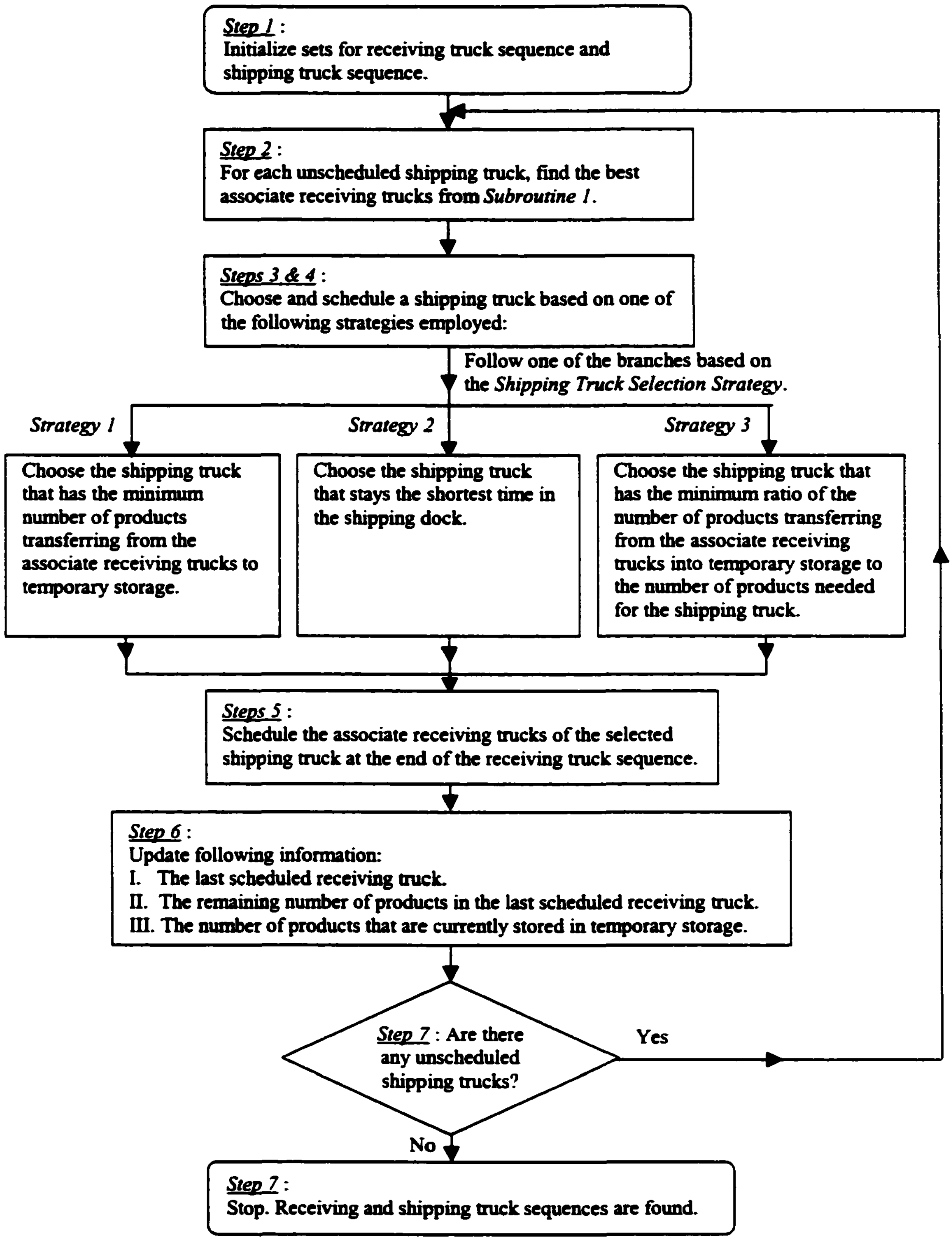


Figure 6. Heuristic Algorithms for the *Case 1* Problem

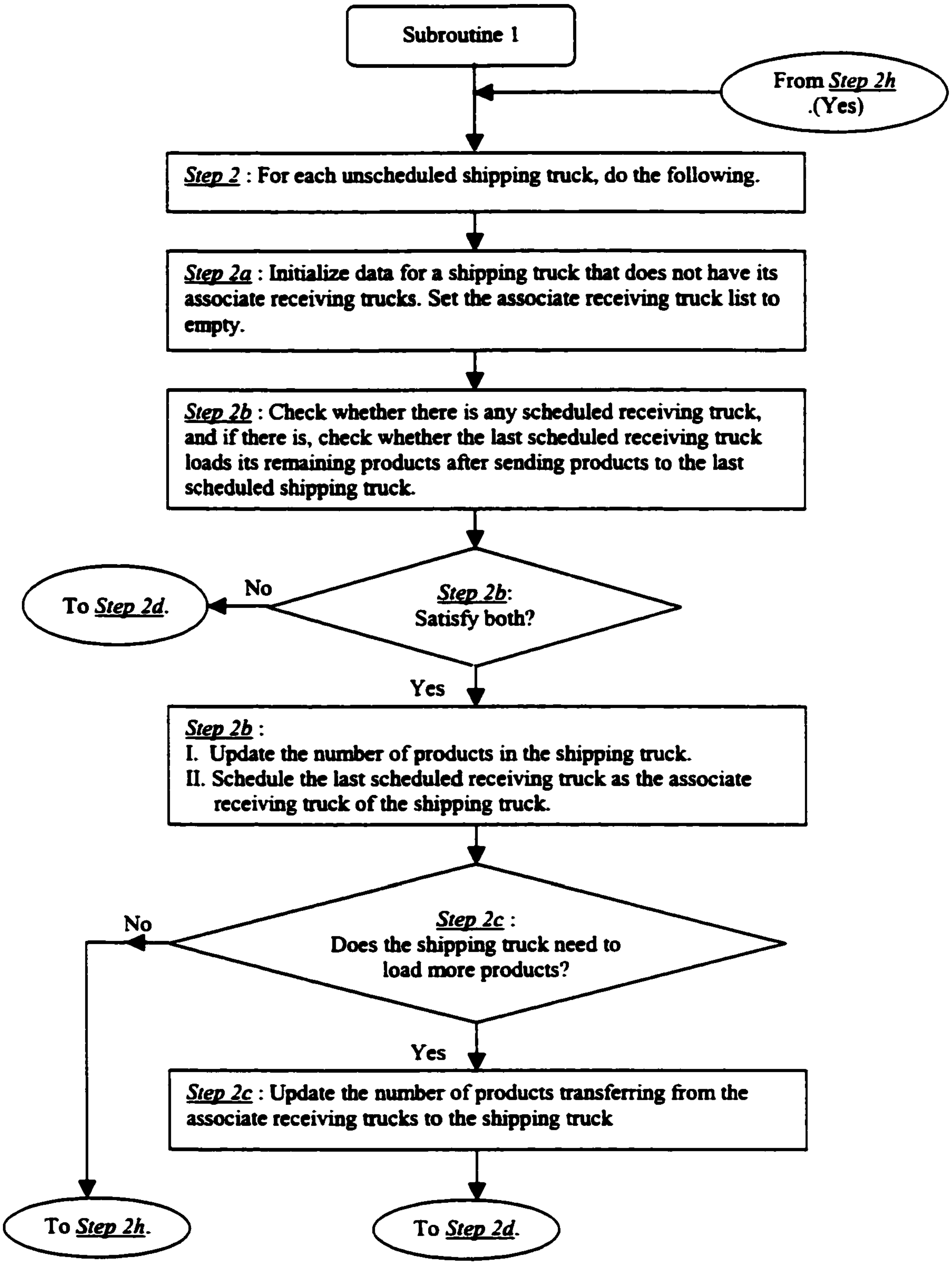


Figure 7. *Subroutine 1* of the Heuristic Algorithms for the *Case 1* Problem

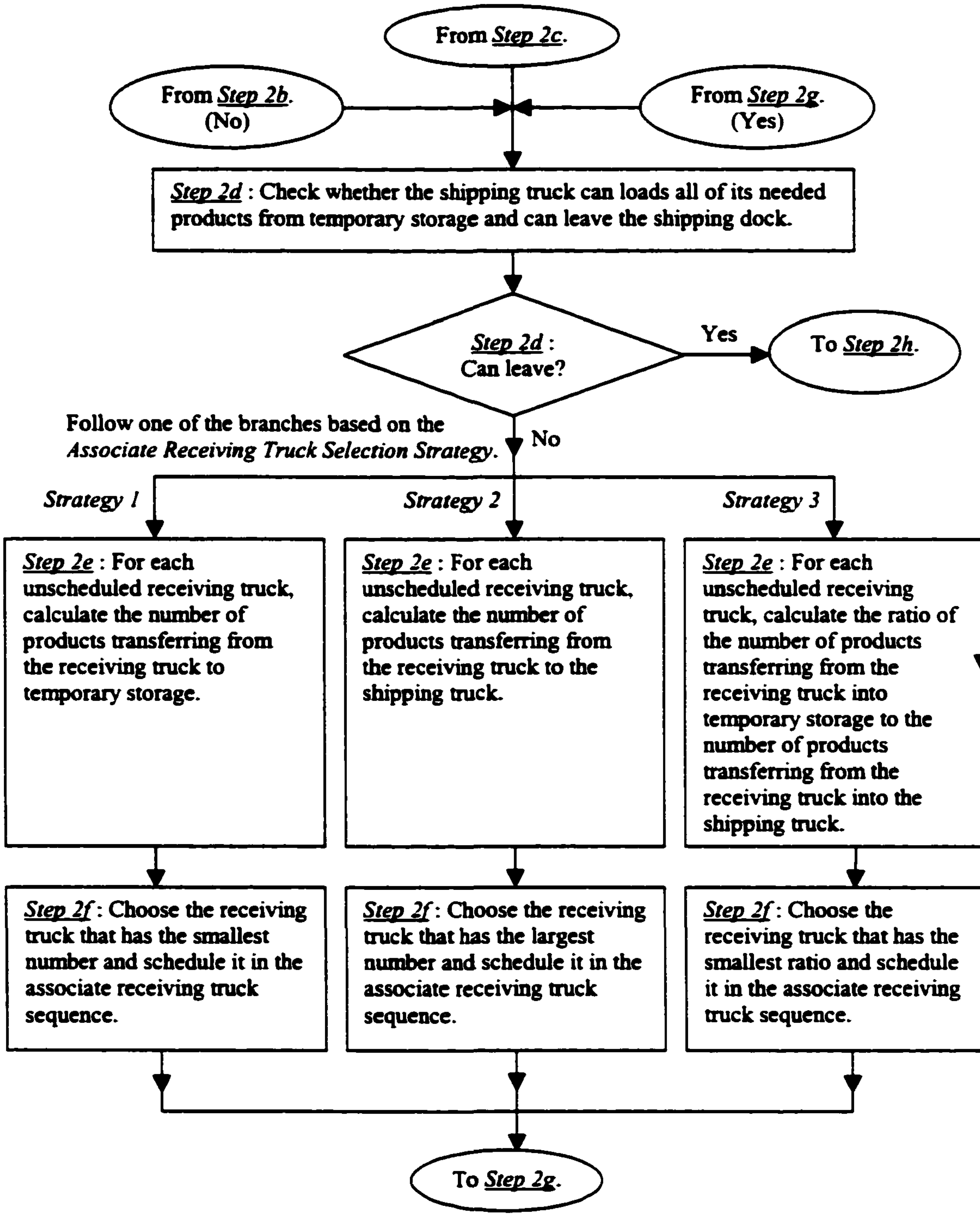


Figure 7. (continued)

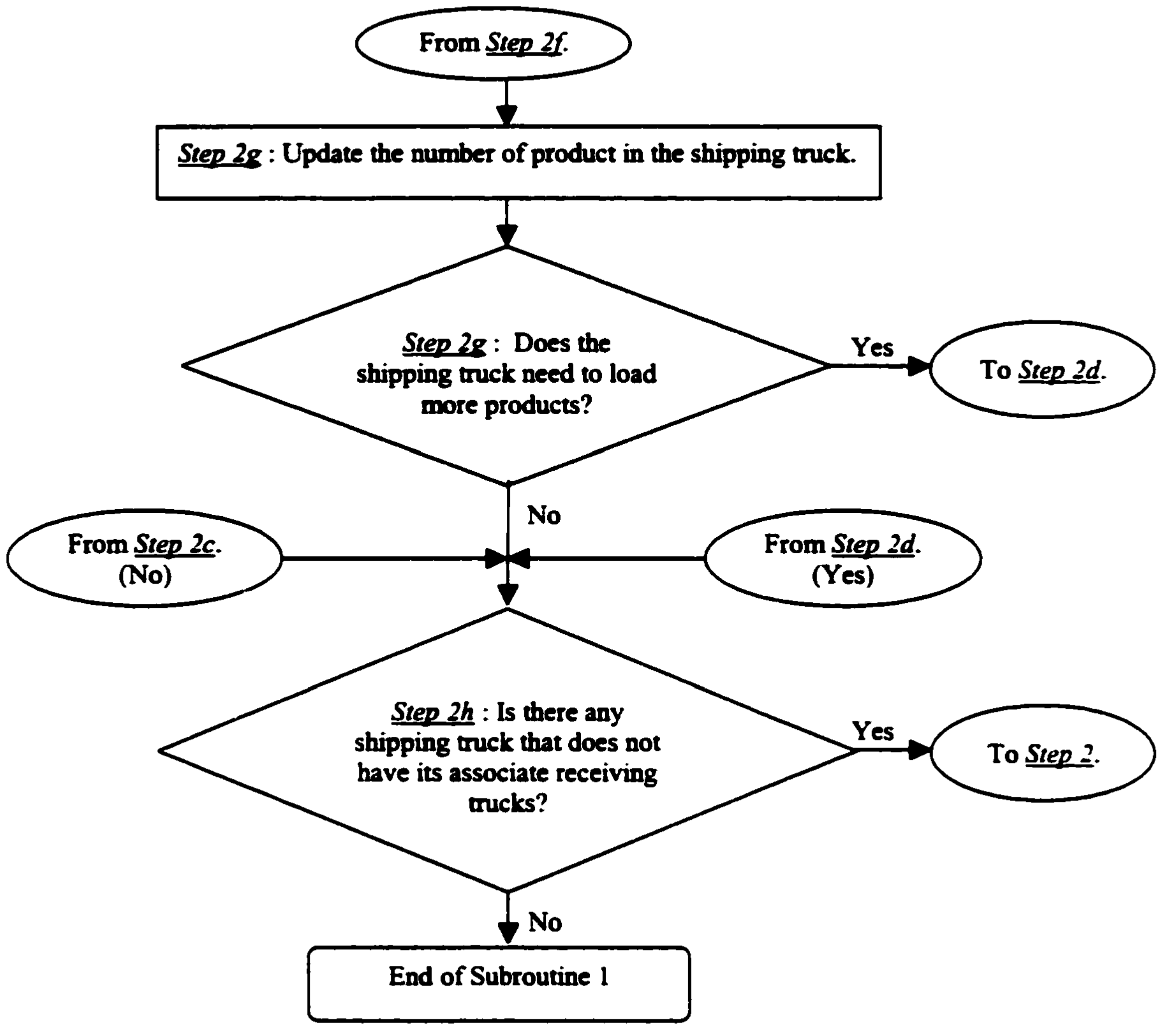


Figure 7. (continued)

4.2.3.5 Different Loading and Unloading Times

So far, it has been assumed that all unloading times and loading times are the same for all products. Now, suppose unloading times and loading times for each type of products are different from one another. Then, the heuristic algorithm can be applied for this situation with a little modification. The major concerns, so far, is to minimize the total number of products that pass through temporary storage. If it is assumed that loading time and unloading time are different for each type of product, the objective will be changed to minimizing the total time required for products that pass through temporary storage.

For this situation, the associate receiving truck selection strategy is slightly modified as follows:

I. *Modified Associate Receiving Truck Selection Strategy 1*

- For each pair of unscheduled shipping truck and the matched or paired receiving truck, the total unloading and loading time required for products that pass through temporary storage is calculated.
- Then, the receiving truck that has the smallest total time required is chosen.

II. *Modified Associate Receiving Truck Selection Strategy 2*

- For each pair of unscheduled shipping truck and the matched or paired receiving truck, the total unloading and loading time required for products that transfer directly from the receiving truck to the shipping truck is calculated.
- Then, the receiving truck that has the largest total time required is chosen.

III. *Modified Associate Receiving Truck Selection Strategy 3*

- For each pair of unscheduled shipping truck and the matched or paired receiving truck, the ratio of the total time required to move products through temporary storage as obtained from *Modified Strategy 1* to the total time required to transfer products directly from receiving truck to shipping truck as obtained from *Modified Strategy 2* is calculated.
- Then, the receiving truck that has the smallest ratio is chosen.

The shipping truck selection strategy is slightly modified as follows:

I. *Modified Shipping Truck Selection Strategy 1*

- For each unscheduled shipping truck and its associate receiving trucks, the total unloading and loading time required for products that pass through temporary storage is calculated.
- Then, the shipping truck and its associate receiving trucks that have the smallest total time required are chosen.

II. *Modified Shipping Truck Selection Strategy 2*

- The shipping truck that stays the shortest time in the shipping dock and its associate receiving trucks are chosen.

III. *Modified Shipping Truck Selection Strategy 3*

- For each unscheduled shipping truck and its associate receiving trucks, the total unloading and loading time required for products that pass through temporary storage is calculated. The total unloading and loading time required for products that transfer directly from the associate receiving trucks to the shipping truck is also calculated. Then, the ratio of total time required for products that pass through temporary storage to the total time required for products that transfer directly from the associate receiving trucks into the shipping truck is calculated.
- The shipping truck and its associate receiving trucks that have the smallest ratio are chosen.

Equations (4-14) to *(4-20)* can be easily modified to determine the amount of time spent instead of the number of products. The appropriate time factors such as u_k, l_k, u^t_k or l^t_k can be multiplied to the number of products in order to calculate the amount of time spent, where time factors u_k, l_k, u^t_k and l^t_k are defined as follows:

u_k = Unloading time for one unit of product k from a receiving truck onto the dock,

l_k = Loading time for one unit of product k from the dock to a shipping truck,

u^t_k = Unloading time for one unit of product k from conveyor to temporary storage,

l^t_k = Loading time for one unit of product k from temporary storage to a shipping truck.

4.2.4 Tabu Search

The tabu algorithm is a search technique aimed at building extended neighborhood procedures, with particular emphasis on avoiding being caught in a local optimum. Similar to simulated annealing and genetic algorithm, the tabu search method has been widely used to solve problems of practical sizes in recent years.

4.2.4.1 General Concepts of Tabu Search

The basic form of the tabu search is founded on ideas proposed by Fred Glover. The method is based on procedures designed to cross boundaries of feasibility or local optimality, which were usually treated as barriers. A tabu search is a meta-heuristic technique that guides a local heuristic search procedure to explore the solution space beyond local optimality. The local procedure is a search that uses an operation called 'move' to define the neighborhood of any given solution. One of the main components of the tabu search is its use of adaptive memory, which creates a more flexible search behavior. Memory-based strategies are therefore the hallmark of tabu search approaches (Glover, 1997).

The idea of the tabu search is quite simple. In a tabu search, the best move available is always taken, even if this makes the objective value somewhat worse. This is basically a diversification move, because intensification is momentarily of no advantage. Now, if the move gets out of the local optimum on the very next move, the objective can probably be decreased the most by moving right back to the same local optimum. Therefore, the search has to be forced to continue diversifying for a few moves. The approach that the tabu search employs to prevent returning to the same local optimum is to keep a list of the last m moves and not to allow moves in the list to be repeated while they remain on the list (they are currently "tabu"). Glover gave the number of moves, m, in the list typically to be set equal to 7 (i.e., $m = 7$).

Before reaching the local optimum, the neighborhood procedure will improve at each step, so that no repetition is possible and the current m moves, which are "tabu" would never be chosen anyway. However, after leaving the local optimum and attempting to try to diversify into a different region of solutions, the tabu list hopefully forces diversification until the old solution area is left behind.

A tabu search has the following characteristics:

(a) The local optimum trial solution can be saved as the best to date, so that nothing is lost by continuing.

(b) The main extra work in the procedure is in keeping an updated list of the last *m* solution sequences and in checking whether each proposed new step is "tabu".

(c) The procedure will not stop even if, in fact, the global solution has been found; there must be some other termination procedure even if it is only the number of moves or elapsed time.

(d) Since a tabu search requires the best move at each choice point, rather than simply an improved one (which may not exist), the selection of a move may become very expensive for very large problems if the neighborhood (number of possible next choices) of a move is very large and/or if the computation involved in evaluating each interchange is very large.

(e) There is the idea of an *aspiration criterion* in tabu search. If a solution is the best solution found to date or it is interesting for some reason, there is no justification to make the solution tabu. So one or more aspiration criteria can be defined that are used to overrule the tabu criteria. If a solution satisfies the aspiration criterion, it is exempted from being tabu. That is, exploring new directions out of this point is more interested than the point itself.

A tabu search is the modern form of an extended neighborhood search. Neighborhood search is a general purpose heuristic technique, which can provide results often very close to optimal at a practical computational cost. The basic elements of the neighborhood search procedure are:

(a) A starting solution to the problem of interest - the *original seed*.

(b) All solutions "close to" the original solution - the *neighborhood of the seed*.

(c) A method for selecting the new seed (improved solution) - the *selection criterion*.

(d) A method for terminating the procedure - the *termination criterion*.

The neighborhood search procedure may be used for quite complicated problems where a solution is itself very complex.

4.2.4.2 Tabu Search applied to the *Case 1* Problem

Similar to the heuristic algorithm for the *Case 1* problem presented earlier, tabu search for the *Case 1* problem was developed to minimize the number of products that pass through temporary storage. Therefore, the solution of the tabu search is presented as the number of products that pass through temporary storage. Tabu search for the *Case 1* problem used the following basic elements of the neighborhood search procedure:

1. *Initial Seed* - There are a number of ways of obtaining the initial seed. The initial seed is randomly picked for the *Case 1* problem.

2. *Neighborhood of the Current Solution* - The adjacent pairwise interchange operation is used to generate a neighborhood of a current solution. Suppose the current receiving truck sequence and shipping truck sequence are scheduled as follows:

   $$T^r = \{t^r_{[1]}, t^r_{[2]}, t^r_{[3]}, \ldots, t^r_{[R-1]}, t^r_{[R]}\} \ \& \ T^s = \{t^s_{[1]}, t^s_{[2]}, t^s_{[3]}, \ldots, t^s_{[S-1]}, t^s_{[S]}\}.$$

   Then, the neighborhood of the current solution would be exactly the following $(R+S-2)$ sequences:

   $$T^r = \{t^r_{[2]}, t^r_{[1]}, t^r_{[3]}, \ldots\ldots, t^r_{[R-1]}, t^r_{[R]}\} \ \& \ T^s = \{t^s_{[1]}, t^s_{[2]}, t^s_{[3]}, \ldots, t^s_{[S-1]}, t^s_{[S]}\}.$$

   (Interchange receiving trucks $t^r_{[1]}$ and $t^r_{[2]}$).

   $$T^r = \{t^r_{[1]}, t^r_{[3]}, t^r_{[2]}, \ldots\ldots, t^r_{[R-1]}, t^r_{[R]}\} \ \& \ T^s = \{t^s_{[1]}, t^s_{[2]}, t^s_{[3]}, \ldots, t^s_{[S-1]}, t^s_{[S]}\}.$$

   (Interchange receiving trucks $t^r_{[2]}$ and $t^r_{[3]}$).

   o

   o

   $$T^r = \{t^r_{[1]}, t^r_{[2]}, t^r_{[3]}, \ldots\ldots, t^r_{[R]}, t^r_{[R-1]}\} \ \& \ T^s = \{t^s_{[1]}, t^s_{[2]}, t^s_{[3]}, \ldots, t^s_{[S-1]}, t^s_{[S]}\}.$$

   (Interchange receiving trucks $t^r_{[R-1]}$ and $t^r_{[R]}$).

   $$T^r = \{t^r_{[1]}, t^r_{[2]}, t^r_{[3]}, \ldots, t^r_{[R-1]}, t^r_{[R]}\} \ \& \ T^s = \{t^s_{[2]}, t^s_{[1]}, t^s_{[3]}, \ldots, t^s_{[S-1]}, t^s_{[S]}\}.$$

   (Interchange shipping trucks $t^s_{[1]}$ and $t^s_{[2]}$).

   $$T^r = \{t^r_{[1]}, t^r_{[2]}, t^r_{[3]}, \ldots, t^r_{[R-1]}, t^r_{[R]}\} \ \& \ T^s = \{t^s_{[1]}, t^s_{[3]}, t^s_{[2]}, \ldots, t^s_{[S-1]}, t^s_{[S]}\}.$$

   (Interchange shipping trucks $t^s_{[2]}$ and $t^s_{[3]}$).

   o

   o

   $$T^r = \{t^r_{[1]}, t^r_{[2]}, t^r_{[3]}, \ldots, t^r_{[R-1]}, t^r_{[R]}\} \ \& \ T^s = \{t^s_{[1]}, t^s_{[2]}, t^s_{[3]}, \ldots, t^s_{[S]}, t^s_{[S-1]}\}.$$

   (Interchange shipping trucks $t^s_{[S-1]}$ and $t^s_{[S]}$).

Even though this neighborhood limits the number of new choices to search, it is relatively small and easy to generate, so there is a trade-off.

3. *Selection Criterion* - To select the next solution after the adjacent pairwise interchange, all solutions in the neighborhood are evaluated and the best solution among all neighborhood solutions is chosen as the next solution even if this makes the objective function value somewhat worse.

4. *Termination* - Tabu search will stop if there is no improvement of the objective for a maximum number of iterations specified by the user. In other words, if a certain number of consecutive solutions do not improve the current best solution, the algorithm will stop. For the *Case 1* problem, 1000 were used as the maximum number of iterations.

5. *Number of Tabu List* - As Glover suggested, a list seven tabu points were used for this algorithm.

In order to explain the tabu search algorithm for the *Case 1* problem, the following notations are used:

i = Number of iteration,

K = Maximum number of iterations allowed which was set by a user,

T^c = Ordered set of the current receiving and shipping truck sequences,

T' = Best neighborhood of the current receiving and shipping truck sequences,

T^* = Ordered set of the best receiving and shipping truck sequences,

The tabu search algorithm used for the *Case 1* problem is as presented below.

TABU SEARCH ALGORITHM FOR THE *CASE 1* PROBLEM

STEP 1

Generate the initial receiving and shipping truck sequences randomly. Set the current sequence as the initial sequence; (i.e. set T^c).

STEP 2

Select the next receiving and shipping truck sequences as follows:

(2a) For each neighborhood sequence of the current sequence, if there is the same sequence of the neighborhood sequence in the tabu list, do not consider the neighborhood sequence as the next sequence. Otherwise, calculate the total number of products that pass through temporary storage for the neighborhood sequences of the current sequence.

(2b) Among all neighborhood sequences considered in *Step (2a)*, choose the next receiving and shipping truck sequences as the neighborhood sequence that has the smallest total number of products that pass through temporary storage; (i.e. Choose T').

STEP 3

Set the current sequence as the next sequence; (i.e. set $T^c \leftarrow T'$).

STEP 4

If the current sequence is the best solution found so far, set the best sequence as the current sequence (i.e. $T^* \leftarrow T^c$) and set the number of iteration to 1; (i.e. set $i = 1$). Go to *Step 2*. Otherwise, increase the number of iterations by 1; (i.e., set $i \leftarrow i+1$).

STEP 5

If the number of iteration is greater than the maximum number of iteration (i.e. $i > K$), stop. Choose the best sequence as the best solution found so far. The best receiving and shipping truck sequences are found. Otherwise, go to *Step 2*.

Figure 8 describes the algorithmic steps of the tabu search algorithm for the *Case 1* problem.

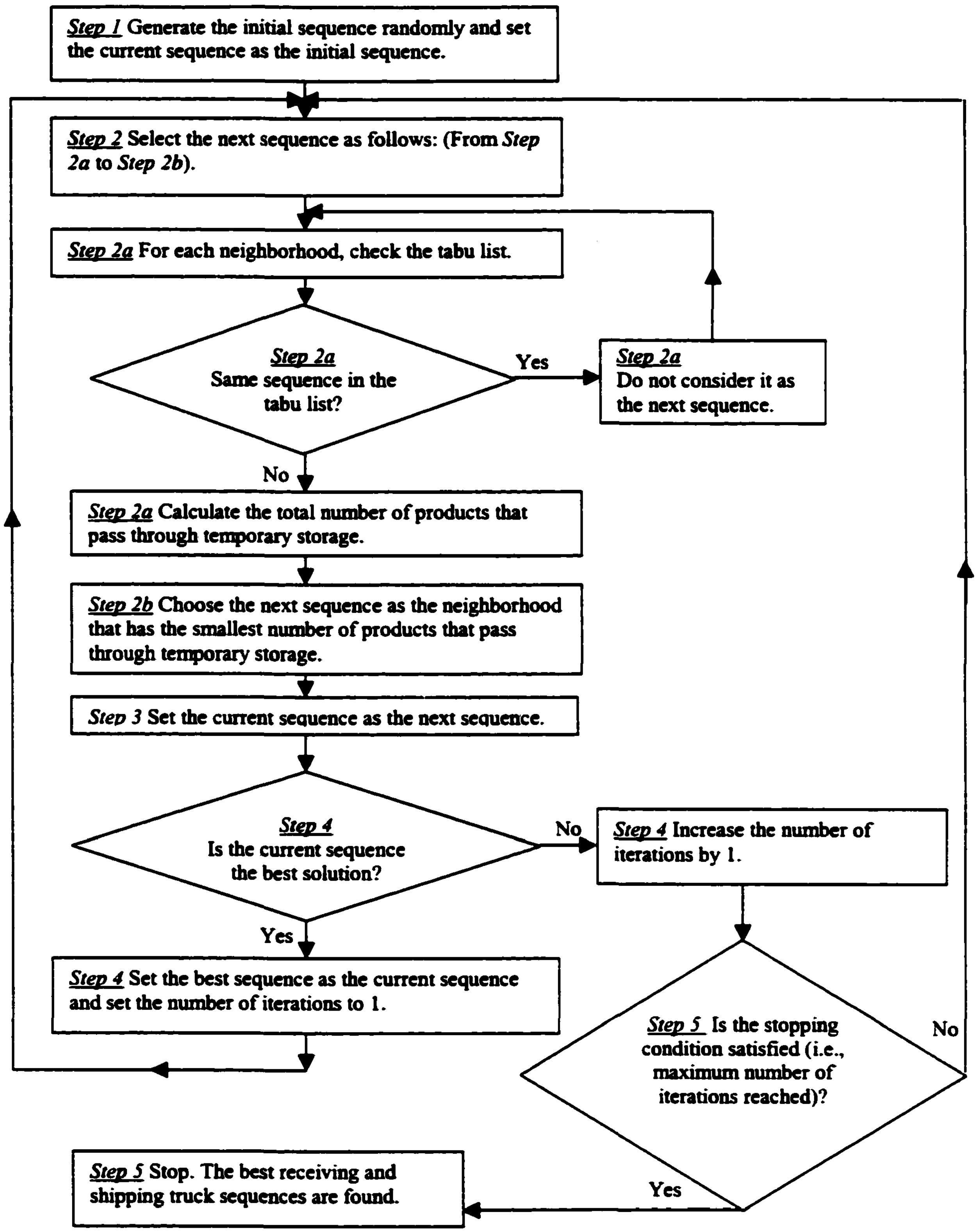


Figure 8. Tabu Search Algorithm for the *Case 1* Problem

4.2.5 Branch and Bound Method

In Section 4.2.2, the complete enumeration approach was used to find the global optimal solution. However, the enumeration approach is not practical to solve reasonable size problems as pointed out in Section 4.2.2. Another solution method used in the research that is able to find the global optimal solution is the branch and bound approach. If a good starting solution is available, the solution can be used as an initial upper bound to the problem in the branch and bound algorithm. With a good starting solution as the initial upper bound, the branch and bound tree should be easily explored, pruned, and completed very quickly.

Branch and bound is a useful method for solving many combinatorial optimization problems. As its name implies, the approach consists of two fundamental procedures. Branching is the process of partitioning a large problem into two or more subproblems, and bounding is the process of calculating a lower bound on the optimal solution of a given subproblem (Baker, 1974).

The branching procedure replaces an original problem by a set of new problems that are

(i) mutually exclusive and exhaustive subproblems of the original problem,
(ii) partially solves versions of the original problem, and
(iii) smaller problems than the original.

Furthermore, the subproblems can themselves be partitioned in a similar fashion.

The basic idea of branching is to conceptualize the problem as a decision tree. From each decision point, called a node, for a partially completed solution, there grows a number of new branches. Each branch makes each possible decision point. These in turn become new nodes for branching again, and so on. Leaf nodes, which cannot be branched from any further represent complete solutions. If the solutions for all the leaf nodes are found, the lowest of these will be the optimal solution. Of course, this will be too expensive except for very small problems. This is where the bounding procedure comes in.

Suppose that at some intermediate stage a complete solution has been obtained that has a performance measure π. Suppose also that a subproblem encountered in the branching process has a lower bound $\sigma > \pi$. Then that subproblem need not be considered any further in the search for an optimum; all branches sprouting from this node and their descendants from the tree can be pruned. When a subproblem (equivalently a node) is found not to be worthy of further branching, such a node is said to be fathomed. By not branching any further from fathomed nodes, the enumeration process is curtailed (Morton, 1993).

For the *Case 1* problem, the branch and bound method is applied to the original problem using the best solution from the heuristic algorithm as the initial upper bound. In each node, the total number of products passing through the temporary storage is calculated. In the typical scheduling problem, only one schedule needs to be made for the problem. However, in the cross docking problem, two sequences need to be made; one is the receiving truck sequence and the other is the shipping truck sequence. Therefore, the typical branch and bound method needs to be modified to construct two sequences.

For the *Case 1* problem, the following branch and bound method is suggested.

1. Take the current best solution, π, from the heuristic algorithm as an upper bound to the problem.
2. Next, solve the original problem using branch and bound and compute the total number of products, σ, that pass through the temporary storage at each node based on the solution up to that node.
3. Compare the total number of products passing through the temporary storage, σ, with the current best solution, π.
   a) If σ is equal to or greater than the current best solution, fathom the node.
   b) If the leaf node is reached and σ is equal to or greater than the current best solution, fathom the node.
   c) If the node is a leaf node and σ is lower than the current best solution, update the current best solution, $\pi \leftarrow \sigma$.
4. Continue until all nodes are fathomed.

The nodes indicate the trucks being scheduled. The sequence of the nodes indicates the order in which the trucks are scheduled at the docks. The following branching strategy was developed for the *Case 1* problem.

1. The first node branched from the root node is used for the shipping trucks.
2. If all products for the last scheduled shipping truck are filled from the scheduled receiving trucks, the next branch will be used for a shipping truck. Otherwise, the next

branch will be used for a receiving truck. This strategy will be explained later with an example.

The suggested branch and bound method is applied to an example problem referred to as Example 2. The problem has three receiving trucks, four shipping trucks and nine product types. Information about each receiving truck and shipping truck is presented in Table 4. It is assumed that all loading and unloading times for all products are the same and are one unit of time in duration.

Table 4 is the same test set as *Test Set 10* in Appendix B. The optimal solution for this problem was 155 products obtained from the complete enumeration method. (The total number of products that passed through the temporary storage is 155.) The optimal sequence for the receiving trucks is 1→3→2 and the optimal sequence for the shipping trucks is 4→1→3→2. The heuristic algorithm also found the same optimal solution.

Table 4. Example Set 2 to Illustrate the Branch and Bound Method for the *Case 1* Problem

Receiving Truck

Truck	Product	Quantity
1	1	64
	2	58
	3	19
	4	38
	5	19
	6	58
	7	19
	8	7
	9	58
2	5	132
	9	118
3	2	49
	7	97
	8	49
	9	145

Shipping Truck

Truck	Product	Quantity
1	4	38
	6	29
	8	28
2	5	151
	6	10
	7	12
	8	28
3	2	41
	6	19
	7	61
	9	229
4	1	64
	2	66
	3	19
	7	43
	9	92

After applying the branch and bound method to Example 2, the solution was obtained as presented in Figure 9. Figure 9 illustrates how the suggested branch and bound method works. From the figure, the optimal path is obtained as follows:

Root→S4→R1→R3→S1→S3→R2→S2.

Suppose node *R1* as shown in *(a)* of Figure 10 needs to be branched from at a given iteration. From the branch and bound tree, it can be seen that receiving truck 1 and shipping truck 4 are scheduled up to this point. At this point, the shipping truck 4 does not load all of its needed products yet. In other word, the shipping truck 4 needs to load more products after it loads products from receiving truck 1. Therefore, the tree will branch out next on the remaining receiving trucks again as shown in *(b)* of Figure 10. Now, suppose node *R3* as shown in *(c)* of Figure 10 needs to be branched in a given iteration. Then, the shipping truck 4 is ready to leave the shipping dock because it has loaded all of its needed products from receiving truck 1 and receiving truck 3. Therefore, the system will branch out next on the remaining shipping trucks as shown in *(d)* of Figure 10. The remainder of the branching process continues in a similar fashion until all nodes are fathomed and the optimal solution is found.

As can be seen in Figure 9, the amount of calculations for a branch and bound method can decrease dramatically by starting with a good upper bound obtained from the heuristic algorithm. For the problem in Example 2, the entire problem was completely enumerated implicitly in 31 nodes by the branch and bound method. If the exhaustive enumeration method is used, the total possible number of combinations for receiving and shipping trucks is 144, which is much larger than 31. For a small problem, it is possible that the amount of calculations required for the branch and bound method is larger than the amount of calculations required for the exhaustive enumeration method. This is because the total number of nodes in the branch and bound tree is larger than the total number of possible truck sequences or leaf nodes in the branch and bound tree. However, the amount of calculations for the branch and bound technique is generally smaller than the amount of calculations required by the enumeration method, if a good initial upper bound is used as shown in Example 2. Therefore, the branch and bound method can be used to find the optimal solution for moderate size problems.

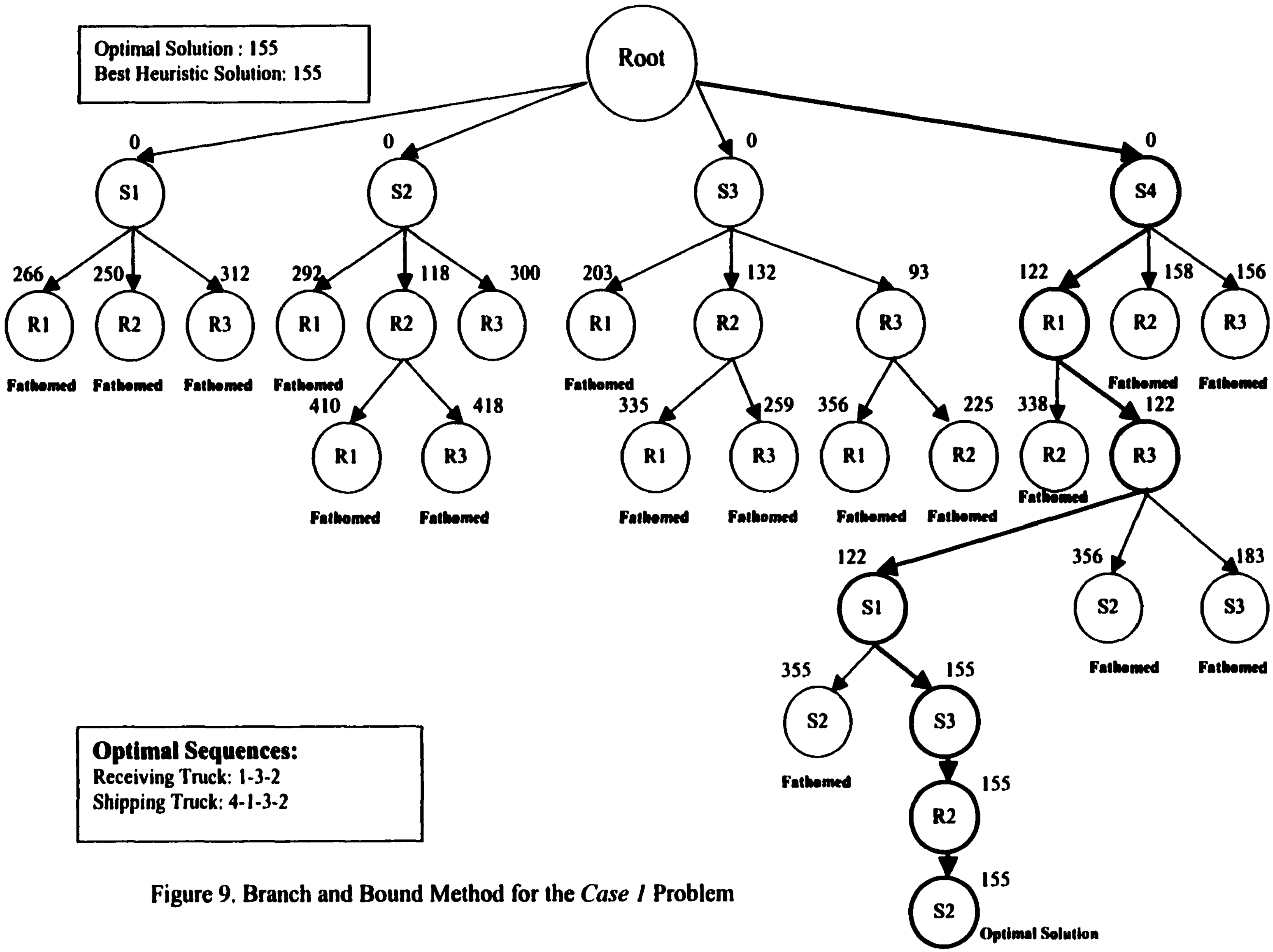


Figure 9. Branch and Bound Method for the *Case 1* Problem

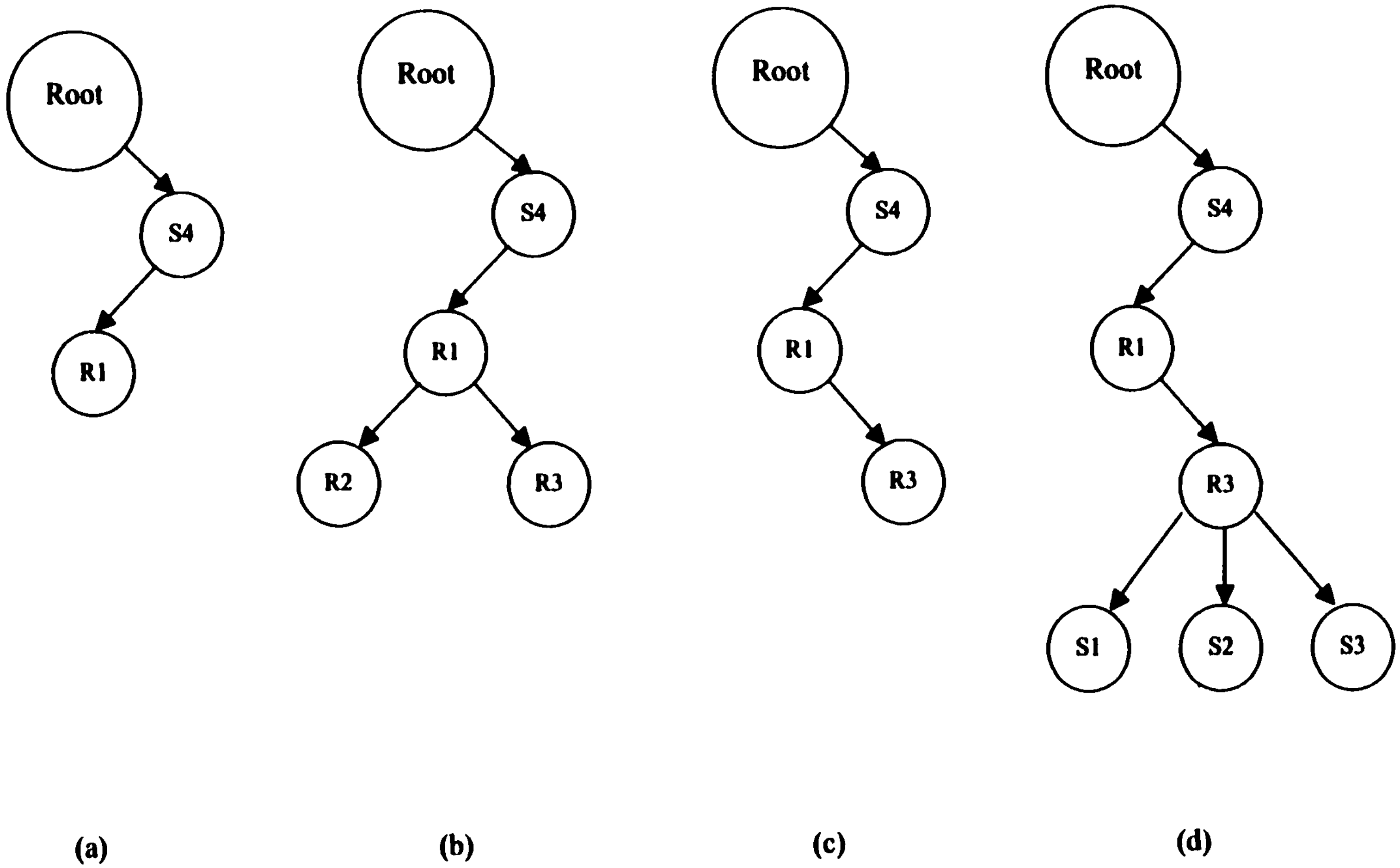


Figure 10. Explanation of Branching Strategy for the *Case I* Problem

4.2.6 Makespan for the *Case 1* Problem

From the solution of the *Case 1* problem, the following information is known: the receiving truck sequence, the shipping truck sequence, the product routing, and the total number of products transferred from a receiving truck to a shipping truck. With the above information, the makespan of a cross docking operation can be found. Unfortunately, the makespan cannot be expressed in a single equation because of the dynamic characteristics of a cross docking operation. Therefore, a procedure was developed to find the makespan of a cross docking operation for the *Case 1* problem. The procedure first finds the departure time of the receiving trucks in the receiving truck sequence, and then the departure times of the shipping trucks in the shipping truck sequence. Throughout this section, it is assumed that all unloading times and loading times are the same for all products and is one unit of time.

4.2.6.1 Notations

To find the makespan for the *Case 1* problem, the following notations are used:

Time:

T = Makespan,

$F_{[i]}$ = Time at which the i^{th} positioned receiving truck in the receiving truck sequence leaves the receiving dock,

$L_{[j]}$ = Time at which the j^{th} positioned shipping truck in the shipping truck sequence leaves the shipping dock,

Data:

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N = Number of product types in the set,

D = Delay time for truck change,

V = Moving time of products from the receiving dock to the shipping dock,

$r_{[i]k}$ = Number of units of product type k initially loaded in the i^{th} positioned receiving truck in the receiving truck sequence,

$s_{[j]k}$ = Number of units of product type k initially needed for the j^{th} positioned shipping truck in the shipping truck sequence,

$t_{[i][j]}$ = Total number of units of product type k which transfers from the i^{th} positioned receiving truck in the receiving truck sequence to the j^{th} positioned shipping truck in the shipping truck sequence,

$$v_{[i][j]} = \begin{cases} 1, & \text{If } t_{[i][j]} > 0 \\ 0, & \text{Otherwise} \end{cases} .$$

4.2.6.2. Calculation of Makespan for the *Case 1* Problem

Suppose that the solution is presented as follows, where $[i]$ represents the i^{th} sequence position in a set rather than the number i itself.

$$T^r = \{ \quad t^r_{[1]},\ t^r_{[2]}, \quad \ldots, \quad t^r_{[R-1]},\ t^r_{[R]} \quad \}.$$
$$T^s = \{ \quad t^s_{[1]},\ t^s_{[2]}, \quad \ldots, \quad t^s_{[S-1]},\ t^s_{[S]} \quad \}.$$

Then makespan is calculated by finding the departure times of the receiving and shipping trucks. The procedure for calculating makespan is presented below.

1. Find the departure time of the receiving trucks corresponding to the receiving truck sequence as follows (i.e. find $F_{[i]}$):
   i) For the departure time, $F_{[1]}$, of the first scheduled receiving truck $i = 1$.

   $$F_{[1]} = \sum_{k=1}^{N} r_{[i]k} \quad . \qquad (4\text{-}21)$$

   The departure time of the first scheduled receiving truck, $F_{[1]}$, is simply the same as the total unloading time required for receiving truck $t^r_{[1]}$.

   ii) For the departure time of the second scheduled receiving truck, $F_{[2]}$, to the last scheduled receiving truck, $F_{[R]}$.

   $$F_{[i]} = F_{[i-1]} + D + \sum_{k=1}^{N} r_{[i]k} \quad . \qquad for\ 2 < i < R \qquad (4\text{-}22)$$

   The departure time of the second scheduled receiving truck to the last scheduled receiving truck is the same as the sum of the departure time of the previously scheduled receiving truck (i.e. $F_{[i-1]}$), the delay time for the receiving truck change, and the total unloading time required for receiving truck $t^r_{[i]}$.

2. Find the departure time of the shipping trucks corresponding to the shipping truck sequence (i.e. find $L_{[j]}$).

   For the departure time of the j^{th} scheduled shipping truck, $L_{[j]}$.

$$L_{[j]} = \max\ \{\lambda_1\ ,\ \lambda_2\}. \tag{4-23}$$

   where,

$$\lambda_1 = \max_{1\le i\le R}\left\{v_{[i][j]}\left(F_{[i]} - \sum_{k=1}^{N} r_{[i]k} + \sum_{k=1}^{N} t_{[i][j]} + V\right)\right\}, \tag{4-24}$$

$$\lambda_2 = L_{[j-1]} + D + \sum_{k=1}^{N} s_{[j]k}\ . \tag{4-25}$$

   Equation (4-23) implies that the departure time of the shipping truck is the same as the larger value of two components λ_1 and λ_2. In the first component λ_1, the term, $F_{[i]} - \sum_{k=1}^{N} r_{[i]k}$, represents the time at which the i^{th} positioned receiving truck in the receiving truck sequence enters into the receiving dock. Therefore, the first component λ_1 represents the time at which the j^{th} positioned shipping truck finishes loading products transferring from the last scheduled receiving truck among receiving trucks which transfer products to the j^{th} positioned shipping truck. The second component λ_2 is the sum of the departure time of the previously scheduled shipping truck $t^s_{[i-1]}$, delay time for a shipping truck change, and the loading time of all needed products for shipping truck $t^s_{[j]}$.

3. After the departure times for all the shipping trucks are found, makespan is $L_{[S]}$ which is the departure time of the last scheduled shipping truck or as given below in equation (4-26).

$$T = L_{[S]}. \tag{4-26-a}$$

$$T = \max_{1\le j\le S}\ \{\ L_{[j]}\ \}. \tag{4-26-b}$$

If different loading time and unloading time are assumed for each type of products, the appropriate time factors such as u_k, l_k, u^t_k or l^t_k can be multiplied to the number of products in order to calculate makespan. The notations u_k, l_k, u^t_k and l^t_k are defined in section 4.2.3.5.

4.3 Implementation and Results

Twenty sets of test problems were randomly generated to test the performances of the heuristic algorithms. The test problems are presented in Appendix B. The first ten sets are the smallest test sets. The range for the number of receiving trucks and shipping trucks is three to five, respectively. Total numbers of products in the set are between 890 and 1030 units. The next ten sets are of moderate size. The number of receiving trucks and shipping trucks are from four to six, respectively. The range for the number of products is 1180 to 2030 units.

After applying the complete enumeration method and the heuristic method to the twenty test problems, the solutions are obtained and they are as presented in Tables 5 and 8. Each method was applied twice with different objectives; 1) minimizing the number of products that pass through temporary storage, and 2) minimizing makespan of a cross docking operation. Table 5 shows the optimal solutions and the worst solutions found by enumerating over all possible sequences. The solutions were found with the objective of minimizing the number of products that pass through temporary storage. Table 5 also shows the average solution of all possible sequences.

Table 6 presents the solutions obtained from the heuristic algorithms. The same objective as in Table 5 was used. Because there are three associate receiving truck selection strategies and three shipping truck selection strategies, a total of nine combinations of strategies are possible. From here, the following notations will be used for convenience:

RS1 - Associate receiving truck selection strategy 1.

RS2 - Associate receiving truck selection strategy 2.

RS3 - Associate receiving truck selection strategy 3.

SS1 - Shipping truck selection strategy 1.

SS2 - Shipping truck selection strategy 2.

SS3 - Shipping truck selection strategy 3.

Table 6 shows solutions for all the nine algorithms as well as the compound heuristic solution of the nine algorithms. The compound heuristic solution is defined as the best solution found after applying all nine heuristic algorithms. For the first ten test problem sets, the heuristic algorithm found the optimal solution in six problem sets. For the last ten test problems, the heuristic algorithm found the optimal solution in three problem sets.

Table 5. Number of Products passing through Temporary Storage founded by Searching All Possible Sequence Combinations for the *Case 1* Problem

Set	Number of Receiving Truck	Number of Shipping Truck	Number of Product Type	Possible Combinations	Optimal Solution	Optimal Sequence	Worst Solution	Worst Sequence	Average
1	4	5	4	2880	93	R: 2-4-1-3 S: 4-3-5-1-2	645	R: 2-3-4-1 S: 5-1-2-3-4	421.54
2	5	4	6	2880	147	R: 1-4-3-5-2 S: 3-2-1-4	772	R: 2-3-4-5-1 S: 3-1-2-4	417.95
3	3	3	8	36	233	R: 3-1-2 S: 1-3-2	461	R: 1-3-2 S: 2-1-3	356.53
4	5	5	8	14400	265	R: 1-3-2-4-5 S: 2-5-4-1-3	792	R: 1-2-3-5-4 S: 1-2-3-4-5	556.28
5	5	3	8	720	180	R: 3-5-1-4-2 S: 1-2-3	546	R: 1-2-3-5-4 S: 2-1-3	446.54
6	4	4	5	576	151	R: 1-2-4-3 S: 1-4-2-3	789	R: 2-3-4-1 S: 1-2-3-4	430.44
7	5	4	6	2880	127	R: 3-5-1-2-4 S: 2-3-1-4	681	R: 1-3-5-4-2 S: 4-1-2-3	463.60
8	3	5	7	720	241	R: 3-1-2 S: 4-2-3-1-5	553	R: 1-2-3 S: 4-1-2-3-5	419.09
9	4	4	8	576	204	R: 2-3-4-1 S: 1-4-2-3	611	R: 2-3-4-1 S: 3-1-2-4	445.13
10	3	4	9	144	155	R: 1-3-2 S: 4-1-3-2	611	R: 1-3-2 S: 2-1-3-4	408.68

Table 5. (continued)

Set	Number of Receiving Truck	Number of Shipping Truck	Number of Product Type	Possible Combinations	Optimal Solution	Optimal Sequence	Worst Solution	Worst Sequence	Average
11	5	4	6	2880	204	R: 4-2-1-3-5 S: 4-2-1-3	1046	R: 1-2-3-5-4 S: 4-1-2-3	753.10
12	6	4	8	17280	472	R: 3-4-5-6-1-2 S: 2-3-4-1	1488	R: 1-3-4-5-6-2 S: 1-2-3-4	1078.21
13	5	6	8	86400	310	R: 5-4-3-1-2 S: 6-4-1-2-3-5	1266	R: 1-3-4-5-2 S: 5-1-2-3-4-6	923.75
14	5	5	8	14400	305	R: 3-4-1-5-2 S: 4-1-5-2-3	1202	R: 1-3-4-5-2 S: 3-1-2-4-5	838.28
15	6	5	4	86400	219	R: 4-6-5-3-2-1 S: 4-1-5-2-3	1365	R: 1-3-4-5-6-2 S: 5-1-2-3-4	832.69
16	5	6	6	86400	239	R: 2-1-3-5-4 S: 1-3-4-2-6-5	1275	R: 1-3-4-5-2 S: 1-2-3-4-5-6	850.31
17	4	4	7	576	300	R: 4-3-1-2 S: 1-3-2-4	802	R: 1-3-4-2 S: 4-1-2-3	601.99
18	6	6	7	518400	290	R: 2-3-6-5-1-4 S: 4-1-2-6-3-5	1349	R: 1-2-3-4-5-6 S: 6-1-2-3-4-5	1004.81
19	5	5	10	14400	459	R: 3-4-2-1-5 S: 2-4-3-5-1	1216	R: 1-2-4-5-3 S: 2-1-3-4-5	971.98
20	6	6	9	518400	429	R: 1-2-3-5-4-6 S: 1-5-6-3-2-4	1622	R: 1-3-4-5-6-2 S: 5-1-2-3-4-6	1172.73

Table 6. Number of Products passing through Temporary Storage obtained by the Nine Heuristic Algorithms for the *Case I* Problem

Set	Total Number of Products	Exact Solution			Heuristic Solution									
		Optimal	Worst	Average	RS1 SS1	RS1 SS2	RS1 SS3	RS2 SS1	RS2 SS2	RS2 SS3	RS3 SS1	RS3 SS2	RS3 SS3	Compound Solution
1	990	93	645	421.54	133	338	133	133	338	133	133	338	133	133
2	1030	147	772	417.95	147	147	198	207	207	207	207	207	221	147
3	890	233	461	355.86	233	304	298	233	304	298	233	304	298	233
4	1000	265	792	556.28	369	410	354	350	608	319	342	608	354	319
5	960	180	546	446.54	246	298	246	246	298	246	246	298	246	246
6	1020	151	789	430.44	151	297	162	151	297	183	151	297	183	151
7	980	127	681	463.60	170	254	127	170	254	156	170	254	156	127
8	890	241	553	419.09	249	266	379	249	266	379	249	266	379	249
9	900	204	611	445.13	236	274	312	237	262	204	237	262	312	204
10	930	155	611	408.68	155	335	155	155	545	155	155	545	155	155
11	1620	204	1046	753.10	244	264	244	244	204	244	244	204	244	204
12	1950	472	1488	1078.21	485	599	485	636	574	636	599	599	489	485
13	1610	310	1266	923.75	368	859	427	415	768	435	415	771	435	368
14	1680	305	1202	838.28	327	763	327	360	641	444	327	641	327	327
15	2030	219	1365	832.69	225	426	225	253	443	220	225	426	225	220
16	1690	239	1275	850.31	239	488	322	302	411	317	302	411	317	239
17	1180	300	802	601.99	330	614	405	330	586	330	330	586	405	330
18	1770	290	1349	1004.81	388	414	435	388	414	451	388	414	451	388
19	1720	459	1216	971.98	593	569	459	538	552	504	593	569	504	459
20	2020	429	1622	1172.73	609	685	574	523	685	584	609	685	574	523

Of the nine heuristic algorithms, the combination of *RS1* and *SS1* works the best in most cases. As it can be seen, this combination produced the best solutions in 13 out of the 20 problem sets. On the other hand, heuristics involving some combination of *SS2* produced the most inferior solutions in most cases. The reason why strategy *SS2* is considered is that it is similar to the shortest processing time (SPT) rule from the point of view of the shipping truck. The shortest processing time rule is widely used in job scheduling and generally produces good solutions. Heurisitc algorithms with SS2 combinations are not effective in the *Case 1* problem as can be seen in Table 6. One interesting characteristic found from Table 6 is that only the combinations of *RS2* and *SS3* found the best solutions in three problem sets (sets 4, 9 and 15) among nine heuristic algorithms. Similarly, only the combination of *RS1* and *SS3* found the best solutions in two test problems (Sets 7 and 19). It indicates that *SS3* strategy produces good solutions in some cases.

Tables 7 and 8 are similar to Tables 5 and 6, respectively, except that minimizing makespan was used as the objective this time. To calculate makespan, it is assumed that all loading times and unloading times are the same for all types of products and are one time unit in duration. It is also assumed that truck change time takes 75 time units and transferring time of products from the receiving dock to the shipping dock takes 100 time units.

Table 7 shows the optimal solutions, the average solutions and the worst solutions found by enumerating over all possible sequences. One of the important characteristics found from Table 7 is that minimizing the total number of products passing through temporary storage is not equivalent to minimizing makespan. Makespan obtained by minimizing the total number of products passing through temporary storage is the same as makespan obtained by minimizing makespan in only ten test problems out of twenty problems. It implies that makespan can be decreased by sending more products to temporary storage in some cases. The reason why this occurs is due to the dynamic characteristics of a cross docking operation. In a cross docking system, many operations such as unloading operation, loading operation or truck changes can be performed concurrently. For example, if a receiving truck and a shipping truck change at the same time by unloading more products from a receiving truck and sending them to temporary storage, delay time for truck changes may decrease, thus makespan may decrease. However, it must be pointed out that the

Table 7. Makespan founded by Searching All Possible Combinations of Sequences for the *Case 1* Problem

Set	Solution based on the Minimum Number of Products passing through Temporary Storage		Solution based on Makespan					
	Minimum Products	Makespan	Number of Products	Optimal Makespan	Optimal Sequence	Worst Makespan	Worst Sequence	Average Makespan
1	93	1557	97	1557	R: 2-1-3-4 S: 4-5-1-2-3	2260	R: 2-3-4-1 S: 5-1-2-3-4	1923.27
2	147	1577	147	1577	R: 1-4-3-5-2 S: 3-2-1-4	2427	R: 2-3-4-5-1 S: 3-1-2-4	1958.43
3	233	1372	233	1372	R: 3-1-2 S: 1-3-2	1751	R: 1-3-2 S: 2-1-3	1629.94
4	265	1774	298	1749	R: 2-3-1-4-5 S: 2-4-3-1-5	2492	R: 1-2-3-5-4 S: 1-2-3-4-5	2174.84
5	180	1579	288	1579	R: 1-3-5-4-2 S: 1-2-3	2056	R: 1-2-3-5-4 S: 2-1-3	1901.41
6	151	1546	151	1546	R: 1-2-4-3 S: 1-4-2-3	2359	R: 2-3-4-1 S: 1-2-3-4	1934.09
7	127	1653	254	1535	R: 4-2-5-1-3 S: 4-1-3-2	2283	R: 1-2-3-5-4 S: 4-1-2-3	2000.09
8	241	1556	266	1525	R: 3-1-2 S: 2-4-1-3-5	1993	R: 1-2-3 S: 4-1-2-3-5	1832.58
9	204	1532	337	1473	R: 1-2-4-3 S: 3-2-1-4	2061	R: 2-3-4-1 S: 3-1-2-4	1856.87
10	155	1452	155	1452	R: 1-3-2 S: 4-1-3-2	2016	R: 1-3-2 S: 2-1-3-4	1791.56

Table 7. (continued)

Set	Solution based on the Minimum Number of Products passing through Temporary Storage		Solution based on Makespan					
	Minimum Products	Makespan	Number of Products	Optimal Makespan	Optimal Sequence	Worst Makespan	Worst Sequence	Average Makespan
11	204	2270	244	2232	R: 3-5-1-2-4 S: 3-2-1-4	3291	R: 1-2-3-5-4 S: 4-1-2-3	2887.34
12	472	2933	577	2833	R: 1-5-3-4-6-2 S: 3-4-2-1	4138	R: 1-3-4-5-6-2 S: 1-2-3-4	3620.67
13	310	2386	310	2386	R: 5-4-3-1-2 S: 6-4-1-2-3-5	3651	R: 1-3-4-5-2 S: 5-1-2-3-4-6	3204.96
14	305	2484	318	2385	R: 4-5-1-3-2 S: 4-1-5-2-3	3582	R: 1-3-4-5-2 S: 3-1-2-4-5	3072.55
15	219	2745	230	2745	R: 4-6-3-5-2-1 S: 4-1-5-2-3	4170	R: 1-3-4-5-6-2 S: 5-1-2-3-4	3400.12
16	239	2407	294	2407	R: 2-1-3-5-4 S: 1-3-4-2-5-6	3740	R: 1-3-4-5-2 S: 1-2-3-4-5-6	3142.93
17	300	1885	399	1867	R: 3-4-2-1 S: 1-4-3-2	2532	R: 1-3-4-2 S: 4-1-2-3	2278.30
18	290	2502	290	2502	R: 2-3-6-5-1-4 S: 4-1-2-6-3-5	3969	R: 1-2-3-4-5-6 S: 6-1-2-3-4-5	3446.43
19	459	2639	629	2553	R: 3-4-1-2-5 S: 2-3-4-1-5	3636	R: 1-2-4-5-3 S: 2-1-3-4-5	3289.75
20	429	2857	481	2732	R: 1-2-5-3-4-6 S: 1-5-3-6-2-4	4492	R: 1-3-4-5-6-2 S: 5-1-2-3-4-6	3887.25

Table 8. Makespan obtained by the Nine Heuristic Algorithms for the *Case 1* Problem

Set	Exact Solution			Heuristic Solutions									
	Optimal	Worst	Average	RS1 SS1	RS1 SS2	RS1 SS3	RS2 SS1	RS2 SS2	RS2 SS3	RS3 SS1	RS3 SS2	RS3 SS3	Compound Solution
1	1557	2260	1923.27	1569	1772	1569	1569	1772	1569	1569	1772	1569	1569
2	1577	2427	1958.43	1577	1577	1697	1609	1609	1609	1609	1609	1714	1577
3	1372	1751	1629.94	1372	1455	1588	1372	1455	1588	1372	1455	1588	1372
4	1749	2492	2174.84	1838	1880	1898	1840	1963	1932	1789	1963	1898	1789
5	1579	2056	1901.41	1652	1653	1652	1652	1653	1652	1652	1653	1652	1652
6	1546	2359	1934.09	1546	1603	1635	1546	1603	1638	1546	1603	1638	1546
7	1535	2283	2000.09	1625	1535	1653	1625	1535	1671	1625	1535	1671	1535
8	1525	1993	1832.58	1525	1525	1819	1525	1525	1819	1525	1525	1819	1525
9	1473	2061	1856.87	1566	1549	1762	1473	1549	1532	1473	1549	1762	1473
10	1452	2016	1791.56	1452	1487	1452	1452	1846	1452	1452	1846	1452	1452
11	2232	3291	2887.34	2232	2349	2232	2232	2270	2232	2232	2270	2232	2232
12	2833	4138	3620.67	2862	2894	2862	2862	2947	2882	2894	2894	2862	2862
13	2386	3651	3204.96	2490	2735	2737	2562	2925	2745	2562	2969	2745	2490
14	2385	3582	3072.55	2484	2793	2484	2413	2817	2548	2484	2817	2484	2413
15	2745	4170	3400.12	2762	2828	2762	2790	2828	2800	2762	2828	2762	2762
16	2407	3740	3142.93	2407	2662	2443	2415	2550	2460	2415	2550	2460	2407
17	1867	2532	2278.30	1885	1902	2135	1885	1902	1885	1885	1902	2135	1885
18	2502	3969	3446.43	2658	2642	2894	2658	2642	2921	2658	2642	2921	2642
19	2553	3636	3289.75	2719	2687	2639	2719	2687	2640	2719	2687	2640	2639
20	2732	4492	3887.25	3109	3036	3178	3036	3036	3178	3109	3036	3178	3036

differences in makespan between the two different objectives are very small as can be seen in Table 7.

Table 8 shows all solutions for the nine algorithms as well as the compound heuristic solution of the nine algorithms when the objective is to minimize makespan. The results are very similar to Table 6 whose solutions were found with the objective of minimizing the total number of products that pass through temporary storage. For the twenty test problem sets, the compound heuristic algorithm found the optimal solution in nine sets, assuming all the nine strategy combinations are implemented together as part of an integrated solution method. Of the nine heuristic algorithms, the combination of *RS1* and *SS1* works the best in most cases. As can be seen, it found the best solutions in thirteen out of the twenty problem sets. On the other hand, heuristics involving some combinations of *SS2* produced the most inferior solutions in most cases.

Tables 9, 10 and 11 present the comparison between the heuristic algorithms and the optimal solutions to analyze the performances of the heuristic algorithms. Table 9 shows the percentage of total number of products that pass through temporary storage relative to the total number of products in the set. The percentage is calculated as follows:

$$Percentage = \frac{(Total\ number\ of\ products\ passing\ through\ temporary\ stroage)}{(Total\ number\ of\ products\ in\ the\ set)} \times 100 . \tag{4-27}$$

The range of the percentages for the optimal solutions is 9.39%-27.08% for the twenty sets of test problems. For the worst cases of solutions, the range is 51.80%-80.30%. This implies that the number of products that pass through temporary storage can be dramatically decreased if a good schedule for the receiving trucks and shipping trucks spotting is constructed. The range based on the average solutions is 39.98%-58.06%. There is still about 30% difference between the range based on the optimal solutions and the range based on the average solutions. Note that the average solution does not necessarily correspond to the solution of any schedule for a problem set. The compound heuristic algorithms produced solutions whose range is 10.84%-31.90%. This range is very close to the range associated with the optimal solutions.

Table 9. Percentage of Total Number of Products passing through Temporary Storage relative to the Total Number of Products in the Test Set

Set	Total Number of Products	Exact Solutions						Compound Heuristic Solutions	
		Optimal Solutions		Worst Solutions		Average Solutions			
		Products passing through Temporary Storage	%	Products passing through Temporary Storage	%	Products passing through Temporary Storage	%	Products passing through Temporary Storage	%
1	990	93	9.39%	645	65.15%	421.54	42.58%	133	13.43%
2	1030	147	14.27%	772	74.95%	417.95	40.58%	147	14.27%
3	890	233	26.18%	461	51.80%	355.86	39.98%	233	26.18%
4	1000	265	26.50%	792	79.20%	556.28	55.63%	319	31.90%
5	960	180	18.75%	546	56.88%	446.54	46.51%	246	25.63%
6	1020	151	14.80%	789	77.35%	430.44	42.20%	151	14.80%
7	980	127	12.96%	681	69.49%	463.60	47.31%	127	12.96%
8	890	241	27.08%	553	62.13%	419.09	47.09%	249	27.98%
9	900	204	22.67%	611	67.89%	445.13	49.46%	204	22.67%
10	930	155	16.67%	611	65.70%	408.68	43.94%	155	16.67%
11	1620	204	12.59%	1046	64.57%	753.10	46.49%	204	12.59%
12	1950	472	24.21%	1488	76.31%	1078.21	55.29%	485	24.87%
13	1610	310	19.25%	1266	78.63%	923.75	57.38%	368	22.86%
14	1680	305	18.15%	1202	71.55%	838.28	49.90%	327	19.46%
15	2030	219	10.79%	1365	67.24%	832.69	41.02%	220	10.84%
16	1690	239	14.14%	1275	75.44%	850.31	50.31%	239	14.14%
17	1180	300	25.42%	802	67.97%	601.99	51.02%	330	27.97%
18	1770	290	16.38%	1349	76.21%	1004.81	56.77%	388	21.92%
19	1720	459	26.69%	1216	70.70%	971.98	56.51%	459	26.69%
20	2020	429	21.24%	1622	80.30%	1172.73	58.06%	523	25.89%

Table 10. Percentage Performance Difference and Percentage Deviation between the Compound Heuristic Solutions and the Optimal Solutions

Set	Total Number of Products	Optimal Solution	Compound Heuristic Solution	Percentage Difference	Percentage Deviation
1	990	93	133	4.04%	43.01%
2	1030	147	147	0.00%	0.00%
3	890	233	233	0.00%	0.00%
4	1000	265	319	5.40%	20.38%
5	960	180	246	6.88%	36.67%
6	1020	151	151	0.00%	0.00%
7	980	127	127	0.00%	0.00%
8	890	241	249	0.90%	3.32%
9	900	204	204	0.00%	0.00%
10	930	155	155	0.00%	0.00%
11	1620	204	204	0.00%	0.00%
12	1950	472	485	0.67%	2.75%
13	1610	310	368	3.60%	18.71%
14	1680	305	327	1.31%	7.21%
15	2030	219	220	0.05%	0.46%
16	1690	239	239	0.00%	0.00%
17	1180	300	330	2.54%	10.00%
18	1770	290	388	5.54%	33.79%
19	1720	459	459	0.00%	0.00%
20	2020	429	523	4.65%	21.91%

Table 10 presents the percentage performance difference between the compound heuristic solutions and the optimal solutions in terms of total number of products that pass through temporary storage to the total number of products in the set. It is calculated as presented in *equation (4-28)*:

$$\text{Percentage Difference}(\%) = \frac{\begin{pmatrix}\text{Total number of products}\\ \text{passing through temproray}\\ \text{storage for the compound}\\ \text{heuristic solutions}\end{pmatrix} - \begin{pmatrix}\text{Total number of products}\\ \text{passing through temproray}\\ \text{storage for the optimal}\\ \text{solutions}\end{pmatrix}}{\text{Total number of products in the set}} \times 100 \qquad (4\text{-}28)$$

The range in percentage performance difference for the twenty problem sets is 0%-6.88%. The average difference for the twenty sets is 1.78%. This implies the overall performance of the heuristics is within 1.78% from the optimum solution. Therefore, the compound heuristics perform reasonably well.

Table 10 also presents percentage deviation in total number of products that pass through temporary storage as found by the compound heuristic algorithm against the number of products that pass through temporary storage for the optimal solutions. It is calculated as presented in *equation (4-29)*:

$$\begin{pmatrix} \textit{Percentage Deviation} \\ \textit{for Total Number of} \\ \textit{Product Passing Through} \\ \textit{Temporary Storage (\%)} \end{pmatrix} = \frac{\begin{pmatrix} \textit{Total number of products} \\ \textit{passing through temproray} \\ \textit{storage for the compound} \\ \textit{heuristic solutions} \end{pmatrix} - \begin{pmatrix} \textit{Total number of products} \\ \textit{passing through temproray} \\ \textit{storage for the optimal} \\ \textit{solutions} \end{pmatrix}}{\begin{pmatrix} \textit{Total number of products} \\ \textit{passing through temproray} \\ \textit{storage for the optimal} \\ \textit{solutions} \end{pmatrix}} \times 100 \quad (4\text{-}29)$$

In the worst case, 43.01% more products pass through temporary storage for the compound heuristic solution than for the optimal solution. The average is 9.91%. It appears the compound heuristic algorithm performs poorly based on this measure. However, it is worth noting that the range for the percentage deviation between the worst solutions and the optimal solutions is 97.85%-593.55%. The average deviation is 308.17%. For the average solutions of all possible enumeration sequences, the range is 52.73%-353.27%. The average of the average solutions is 179.64%. Therefore, the compound heuristic solutions perform far better than the average solution.

Table 11 shows the percentage deviation of makespan between the optimal solutions and the compound heuristic solutions. Percentage deviation for makespan is calculated as presented in *equation (4-30)*:

$$\begin{pmatrix} \textit{Percentage Deviation} \\ \textit{for Makespan (\%)} \end{pmatrix} = \frac{\begin{pmatrix} \textit{Makespan for Compound} \\ \textit{Heuristic Solution} \end{pmatrix} - \begin{pmatrix} \textit{Makespan for} \\ \textit{Optimal Solution} \end{pmatrix}}{\textit{Makespan for Optimal Solution}} \times 100 \quad (4\text{-}30)$$

Table 11. Percentage Deviation for Makespan between Optimal Solutions and Compound Heuristic Solutions

Set	Makespan for Optimal Solution	Makespan for Compound Heuristic Solution	Percentage Deviation for Makespan
1	1557	1569	0.77%
2	1577	1577	0.00%
3	1372	1372	0.00%
4	1749	1789	2.29%
5	1579	1652	4.62%
6	1546	1546	0.00%
7	1535	1535	0.00%
8	1525	1525	0.00%
9	1473	1473	0.00%
10	1452	1452	0.00%
11	2232	2232	0.00%
12	2833	2862	1.02%
13	2386	2490	4.36%
14	2385	2413	1.17%
15	2745	2762	0.62%
16	2407	2407	0.00%
17	1867	1885	0.96%
18	2502	2642	5.60%
19	2553	2639	3.37%
20	2732	3036	11.13%

The range of percentage deviation for makespan for the twenty problem sets is 0%-11.13%. The overall average percentage deviation for makespan is 1.80%. This shows again that makespans found from the compound heuristic algorithms are very close to the optimal solutions for makespan.

After applying the tabu search to the same twenty sets of test problems, the solutions are obtained and they are as presented in Table 12. Because the tabu search can possibly find different solutions to a given problem based on the initial receiving and shipping truck sequences specified, the tabu search was run ten times for each test problem to test the quality of tabu solution and reduce the chances of obtaining inferior solution or local optimum. Each run employed a different starting solution that was generated randomly.

Table 12 shows the global optimal solutions obtained from the enumeration method, the compound heuristic solutions, the best tabu search solutions, the worst tabu search solutions and the average tabu search solutions of ten runs. The solutions are presented as the total number of products that pass through temporary storage.

When the best tabu search solutions are examined, the tabu search found the optimal solution in twelve out of the twenty test sets. If the best solution out of the ten runs per each problem set is selected, it appears the performance of tabu search is better than that of the compound heuristic algorithm since the compound heuristic algorithm found the optimal solution in only nine problem sets. However, it needs to be emphasized that the tabu search solution is based on the initial receiving and shipping truck sequences and that the best solution indicated is the best of ten solutions with ten different starting sequences. An alternative to using the best solution from tabu for comparison with the compound heuristic algorithm is to use the average tabu solution for each of the test problems.

When the compound heuristic solution and the average tabu search solution are compared, the compound heuristic solution is found to dominate the average tabu search solution in eighteen sets. In other words, the compound heuristic solution found better solutions in eighteen sets. On the other hand, the average tabu search solution is better than the compound heuristic solution in only one problem set which in this case is problem set 1. For problem set 3, the solutions are the same.

At this point, it can be argued that if several initial random sequences are used for the tabu search instead of using one initial random sequence, the performance of the tabu search will be better than the performance of the heuristic algorithm. This implies that as the number of starting solutions is increased, more improved solution would probably be found, but the computational time will increase at the same time. To test the performance of the algorithm using several initial sequences, the tabu search is modified as described in the following paragraphs.

Table 12. Tabu Search Solutions for the *Case 1* Problem

Problem Number	Optimal Solution	Heuritic Solution	Best Tabu Solution	Average Tabu Solution	Worst Tabu Solution
1	93	133	93	115.1	164
2	147	147	165	178.5	203
3	233	233	233	233	233
4	265	319	286	343.1	427
5	180	246	217	247.1	327
6	151	151	151	157.8	219
7	127	127	127	140.3	158
8	241	249	241	266.4	352
9	204	204	204	222.0	265
10	155	155	155	175.8	259
11	204	204	204	242.7	403
12	472	485	472	585.5	870
13	310	368	346	463.9	725
14	305	327	305	388.0	572
15	219	220	256	287.0	349
16	239	239	252	324.6	513
17	300	330	300	360.9	405
18	290	388	290	392.3	522
19	459	459	504	552.8	568
20	429	523	433	524.5	654

The same tabu search algorithm was used except for the *Termination* rule of the tabu search. In the original tabu algorithm, the tabu search will stop if there is no improvement in the objective function value after the maximum number of iterations is executed. In the modified tabu search, the tabu search will stop if there is no improvement in the objective function value after a maximum number of random initial sequences has been employed. The initial stopping criterion was instead used to stop the search associated with a given initial sequence and to start a new random starting sequence while the modified stopping criterion is used to completely halt the search once the maximum number of starting random sequences has been tried out. Therefore, there is one more loop outside the original tabu search to generate the starting random sequences. To test the performance of the algorithm based on the modified termination criterion, 100 was used as the maximum number of random initial sequences. Therefore, up to one hundred initial sequences could be generated to start the search for each set of test problem.

Table 13 shows the performance of the modified tabu search algorithm. As it can be seen, the modified tabu search found the optimal solution in all twenty problem sets. However, computational time increased significantly. All algorithms were implemented on a personal computer (Intel Pentium Pro Microprocessor 200MHz) and the search times were noted. It took about 7 to 41 seconds to run the modified tabu search algorithm. In all cases, the modified tabu search algorithm took more time than even the enumeration method. However, the time required for the enumeration method increased exponentially as the number of receiving and shipping trucks increased. In the modified tabu search, the computational time does not significantly increase with increases in the number of receiving and shipping trucks employed since there is an upper bound of 100 on the number of initial random sequences that can be used. This upper bound is independent of the fleet sizes of the receiving and shipping trucks employed. For the heuristic algorithms, it only took less than 0.01 seconds to solve all nine heuristic algorithms for each test problem set and it still found good solutions.

4.4 Conclusions

To solve the cross docking problem for the *Case 1* model, five different solution approaches were developed. The first approach is a mathematical model whose objective is to minimize the makespan of a cross docking operation. The second approach employed complete enumeration of all possible sequences to find an optimal solution. While for problems of small sizes, the first two approaches can be used, they are inefficient and impractical to use for medium to large size problems because of the increased computational load required to solve the problems. Therefore, to increase solution efficiency, a heuristic algorithm was developed.

Table 13. Modified Tabu Search Solutions for the *Case 1* Problem

Problem Number	Enumeration Method		Heuristic Algorithm		Original Tabu Search		Modified Tabu Search	
	Optimal Solution	Time (Second)	Compound Solution	Time (Second)	Best Solution	Time (Second)	Solution	Time (Second)
1	93	0.098	133	0.004	93	0.240	93	10.966
2	147	0.280	147	0.006	165	0.421	147	17.444
3	233	0.083	233	0.003	233	0.158	233	7.344
4	265	1.126	319	0.009	286	0.579	265	29.613
5	180	0.131	246	0.007	217	0.360	180	14.385
6	151	0.157	151	0.007	151	0.277	151	9.723
7	127	0.267	127	0.006	127	0.626	127	16.755
8	241	0.209	249	0.007	241	0.441	241	13.111
9	204	1.208	204	0.006	204	0.373	204	13.932
10	155	0.210	155	0.005	155	0.292	155	11.876
11	204	0.309	204	0.004	204	0.512	204	16.219
12	472	1.181	485	0.007	472	0.712	472	29.005
13	310	4.522	368	0.007	346	0.833	310	36.274
14	305	0.971	327	0.009	305	0.976	305	28.634
15	219	2.295	220	0.007	256	0.606	219	27.172
16	239	3.994	239	0.007	252	0.777	239	25.987
17	300	0.095	330	0.004	300	0.389	300	13.355
18	290	19.471	388	0.009	290	1.129	290	31.971
19	459	1.747	459	0.006	504	0.819	459	36.446
20	429	23.454	523	0.009	433	1.777	429	40.805

The heuristic algorithm consists of two stages. In the first stage, the associate receiving trucks are found for each unscheduled shipping truck. In the second stage, one of the unscheduled shipping truck and its associate receiving trucks are selected and scheduled. Because there are three strategies for selecting the associate receiving trucks and three strategies for selecting shipping trucks, a total of nine combinations of strategies or rules were tested. Of the nine rule algorithms, the combinations of *RS1* and *SS1* performed the best in most cases. In some cases, *SS3* found the best solution among the strategies. Overall, the compound heuristic algorithm produced solutions that were very close to the global optimal solutions. For example, the range of percentage deviations for makespan in twenty problem sets tested was 0%-11.13%. The average deviation for makespan was 1.80%.

To test the performance of the heuristic algorithm, the tabu search was implemented for the *Case 1* problem. A tabu search is a meta-heuristic algorithm that has been widely used to solve combinatorial problems in recent years. In comparing the compound heuristic solutions with the tabu solutons, it was found that the tabu search outperformed the compound heuristic algorithm if the best out of ten different solutions generated by using ten different random starting solutions for each problem set is selected. However, if only one random starting solution is used for the tabu search for each set of test problem and the final solutions obtained from the one single starting random solution were used for comparison with the compound heuristic solutions, then the compound heuristic algorithm outperformed the tabu search.

Because the solution of the tabu search depends on the random initial receiving and shipping truck sequences specified, a modified tabu search is implemented to improve the performance of the algorithm. In the modified tabu search, a user is required to specify the maximum number of initial random solutions to use in starting the search for each test problem. Depending on what the user specifies, the algorithm is run as many times as specified by the user and each time the algorithm starts with a different initial random solution (i.e., initial receiving and shipping truck sequences) for each test problem. The application of multiple random initial solutions and restarting the algorithm with each initial solution significantly improves the chances that the algorithm will find a better solution. In this study, the number of starting random solutions employed to initiate the modified tabu search for each test problem was 100. With the modified tabu search, the algorithm was able to find the optimal solution in all twenty problem sets. However, the computational time to solve the problems also increased. For a large problem set, the heuristic algorithm can be implemented usefully because it can find very good solutions within a reasonable computational load.

The last solution approach applied to the research was the branch and bound method. In the implementation of the branch and bound method, the best solution obtained for each problem set by the compound heuristic algorithm was used as the initial upper bound for the solutions of the test problems. For the same problem set, the branch and bound method took shorter time to find the optimal solutions than the complete enumeration method. The efficiency of the branch aqnd bound approach is significantly improved with the use of good bounds. The use of the compound solutions obtained from the heuristic as the initial upper bound applied by the branch and bound method significantly enhanced its effectiveness

CHAPTER 5. *CASE 2* – CROSSDOCKING MODEL WITH DOCK REPEAT TRUCK HOLDING PATTERN AND NO TEMPORARY STORAGE

5.1 Model Descriptions

In the *Case 2* model, there is no temporary storage in the warehouse or distribution center. However, both the receiving truck and the shipping truck can move in and out of the dock during their tasks. Therefore, it is possible that a receiving truck unloads some of its products on the receiving dock, moves out, waits and goes into the receiving dock again to unload its remaining products. This sequence can be similarly applied to the shipping truck. However, since there is no temporary storage space available, the conveyor operating from a receiving dock to a shipping dock may need to stop if the shipping truck is not ready when a product arrives at the shipping dock.

The objective of the *Case 2* problem is the same as that of the *Case 1* problem. It is to find the best sequence for truck spotting for both the receiving and shipping trucks to minimize total operation time or to maximize the throughput of the cross docking system. As in the *Case 1* problem, the product routing is also decided simultaneously along with the spotting sequence of the receiving and shipping trucks.

In the *Case 2* model, delay time occurs when the shipping truck changes or when the shipping truck is not loading any products from the shipping dock and waits for its needed products to arrive at the shipping dock. The change of receiving trucks at the receiving dock may cause the waiting of the shipping truck at the shipping dock. Therefore, both types of the delay times for the *Case 2* model are related to truck changes. From the above characteristics of the *Case 2* model, it is evident that the makespan can be minimized if the number of truck changes is minimized. Minimizing the number of truck changes is equivalent to minimizing the number of matching pairs of the receiving trucks and shipping trucks. The receiving truck and the shipping truck are said to be paired if any product moves from the receiving truck to the shipping truck. Consequently, makespan can be minimized if the number of matching pairs of the receiving and shipping trucks is minimized.

In the solution, a receiving truck can be paired with several shipping trucks. Similarly, a shipping truck can also be paired with several receiving trucks. Product requirements for the shipping trucks and the flow conservation for each receiving truck must be satisfied in the solution.

5.2 Model Developments

To solve the cross docking problem for the *Case 2* model, three approaches were developed. For the first approach, a mixed integer programming model was developed with the objective of minimizing makespan of a cross docking operation. However, the use of mixed integer programming is not considered suitable for modeling the problem because of the exponential growth in variables and constraints as the number of receiving trucks, shipping trucks, and products increase.

The second approach also applied mathematical programming model using a different objective function. As mentioned earlier, minimizing the makespan is equivalent to minimizing the number of matching pairs of the receiving and shipping trucks for the *Case 2* model. Therefore, the second integer programming model was developed to minimize the number of matching pairs of the receiving truck and the shipping truck while product requirements are satisfied. By changing the objective of the mathematical model, the number of variables and constraints of the second integer programming model were drastically decreased in comparison with the first mixed integer programming model. Although a much larger size problem can be solved by the second model than the by the first model, a considerable amount of time is required to translate it into computer code. This combined with large computational time requires also renders the reapproach unattractive and ineffective for solving large problems. Appendix B which models a small problem scenario illustrates the point.

Although the two mathematical models are, in principle, able to find the global optimal solutions to the *Case 2* problems, they are in general not practical to use because of the intensity of their computational requirements to solve problems of meaningful sizes. Therefore, a third solution approach was developed to solve meaningful size problems. The third approach employs heuristic algorithms. The algorithms are able to find solutions to problems very quickly but no optimal solution is guaranteed. Six heuristic algorithms were developed and tested for the *Case 2* problem.

5.2.1 Mathematical Model I

For mathematical model I of the *Case 2* problem, it is assumed that unloading time from a receiving truck and loading time into a shipping truck are the same for all products and that it takes one unit of time in duration to unload or load one unit of product. Additionally, it is assumed that all operations can be carried out simultaneously. In other words, unloading operations from a receiving truck, loading operations into a shipping truck, or receiving and shipping truck changes can be carried out at the same time. With the above assumptions, the following mixed integer programming model was developed for the *Case 2* problem with the objective of minimizing makespan of a cross docking operation.

5.2.1.1 Notations

The following notations are used in Model I:

Continuous Variables:

T = Makespan,

U_{ij} = Time at which the variable t_{ij} transferring from receiving truck i to shipping truck j start to unload from receiving truck i onto the receiving dock,

L_{ij} = Time at which the variable t_{ij} transferring from receiving truck i to shipping truck j finished loading from the shipping dock into shipping truck j,

Integer Variables:

x_{ijk} = Number of units of product type k which transfer from receiving truck i to shipping truck j,

t_{ij} = Total number of units of products which transfer from receiving truck i to shipping truck j, where $\left(t_{ij} = \sum_{k=1}^{N} x_{ijk} \right)$,

Binary Variables:

$$v_{ij} = \begin{cases} 1, & \text{If any products transfer from receiving truck } i \text{ to shipping truck } j \\ 0, & \text{Otherwise} \end{cases},$$

$$y_{iji'j'} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ immediately or directly precedes the variable } t_{i'j'} \text{ in the receiving or shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$y_{00i'j'} = \begin{cases} 1, & \text{If the variable } t_{i'j'} \text{ is placed at the first position in the receiving or shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$y_{ij00} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ is placed at the last position in the receiving or shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

Data:

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k which is initially loaded in receiving truck i,

s_{jk} = Number of units of product type k which is initially needed for shipping truck j,

D = Delay time for truck change,

V = Moving time of products from the receiving dock to the shipping dock,

M = Big number.

5.2.1.2 Mixed Integer Programming Model (Model I)

The mixed integer programming model for the *Case 2* problem with the objective of minimizing makespan of a cross docking operation is presented below.

Mixed Integer Programming Model of Model I for the *Case 2* Problem

Min T

Subject to

$$T \geq L_{ij}, \quad for\ all\ i, j \tag{5-1}$$

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \quad for\ all\ i, k \tag{5-2}$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \quad for\ all\ j, k \tag{5-3}$$

$$\sum_{k=1}^{N} x_{ijk} = t_{ij}, \quad \text{for all } i, j \tag{5–4}$$

$$t_{ij} \leq M\, v_{ij}, \quad \text{for all } i, j \tag{5–5}$$

$$v_{ij} = \sum_{i'=1}^{R}\sum_{j'=1}^{S} y_{iji'j'} + y_{ij00}, \quad \text{for all } i, j \tag{5–6}$$

$$v_{i'j'} = \sum_{i=1}^{R}\sum_{j=1}^{S} y_{iji'j'} + y_{00i'j'}, \quad \text{for all } i', j' \tag{5–7}$$

$$\sum_{i'=1}^{R}\sum_{j'=1}^{S} y_{00i'j'} = 1, \tag{5–8}$$

$$\sum_{i=1}^{R}\sum_{j=1}^{S} y_{ij00} = 1, \tag{5–9}$$

$$y_{ijij} = 0, \quad \text{for all } i, j \tag{5–10}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} - M\left(1 - y_{iji'j'}\right), \quad \text{for all } i, j, i', j' \text{ and where } i = i' \tag{5–11–a}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} + D - M\left(1 - y_{iji'j'}\right), \quad \text{for all } i, j, i', j' \text{ and where } i \neq i' \tag{5–11–b}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} - M\left(1 - y_{iji'j'}\right), \quad \text{for all } i, j, i', j' \text{ and where } j = j' \tag{5–12–a}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} + D - M\left(1 - y_{iji'j'}\right), \quad \text{for all } i, j, i', j' \text{ and where } j \neq j' \tag{5–12–b}$$

$$L_{ij} \geq U_{ij} + V + t_{ij}, \quad \text{for all } i, j \tag{5–13}$$

$$U_{i'j'} \geq L_{ij} - V - M\left(1 - y_{iji'j'}\right), \quad \text{for all } i, j, i', j' \text{ and where } i \neq i' \text{ or } j \neq j' \tag{5–14}$$

$$\text{all variables} \geq 0.$$

Constraint (5–1) ensures that makespan is equal to or greater than the time the last product is loaded onto the last scheduled shipping truck. *Constraint (5–2)* ensures that the total number of units of product type k that transfer from receiving truck i to all shipping trucks is exactly the same as the number of units of product type k which was initially loaded in receiving truck i. Similarly, *constraint (5–3)* ensures that the total number of units of product type k that transfer from all receiving trucks to shipping truck j is exactly the same as the number of units of product type k needed for shipping truck j. *Constraint (5–4)* defines the t_{ij} variables which is used in *constraints (5–11)* to *(5–13)* in order to calculate the unloading and loading times. *Constraint (5–5)* just enforces the correct relationship between the t_{ij} variables and the v_{ij} variables.

Constraint (5–6) ensures that only one of the t_{ij} variables can immediately or directly precede another $t_{i'j'}$ variable in the receiving or shipping sequence when $v_{ij} = 1$. *Constraint (5–7)* ensures that only one of the $t_{i'j'}$ variables can immediately or directly follow another t_{ij} variable in the receiving or shipping sequence when $v_{i'j'} = 1$. *Constraint (5–8)* ensures only one of the $t_{i'j'}$ variables can be placed at the first sequence position of the receiving or shipping sequence. *Constraint (5–9)* ensures that only one of the t_{ij} variables can be placed at the last sequence position of the receiving or shipping sequence. *Constraint (5-10)* ensures that there are no consecutive sequences that transfer products from the same receiving truck to the same shipping truck.

Constraint (5–11–a) and *(5–11–b)* make a valid sequence of unloading times of the t_{ij} variables, based on their order. If there is no receiving truck change between consecutive unloading sequence (in case of $i = i'$), *constraint (5–11–a)* is applied. However, if there is a receiving truck change between the consecutive unloading sequences (in case of $i \neq i'$), the delay time for receiving truck change must be considered, thus *constraint (5–11–b)* is applied. Similar to *constraints (5–11–a)* and *(5–11–b)*, *constraints (5–12–a)* and *(5–12–b)* make a valid sequence of loading times of the t_{ij} variables, based on their order. If there is no shipping truck change between the consecutive loading sequences (in case of $j = j'$), *constraint (5–12–a)* is applied. However, if there is a shipping truck change between consecutive loading sequences (in case of $j \neq j'$), the delay time for shipping truck change must be considered, thus *constraint (5–12–b)* is applied. Finally, *constraints (5–13)* and *(5–14)* establish the proper relationship between the unloading time and the loading time of the t_{ij} variables.

The number of decision variables for this integer programming model is *RS(RS+N+6)+1*. The decision variables consist of *RS(RS+3)* of binary variables, *RS(N+1)* of integer variables and *(2RS+1)* of continuous variables. The number of constraints is

3RS(RS+2)+N(R+S)+2 including *RS(3RS+2)* of inequality constraints and *(4RS+RN+SN+2)* of equality constraints.

5.2.1.3 Interpretation of the Solution

First, the receiving and shipping sequences of the t_{ij} variables can be found from the $y_{00i'j'}$, $y_{iji'j'}$, and y_{ij00} variables. From the receiving and shipping sequences of the t_{ij} variables, the receiving truck spotting sequence and the shipping truck spotting sequence can be identified. The product routing can be found from the t_{ij} and x_{ijk} variables. The variable T represents the makespan of the cross docking operation. The detailed information about the unloading time and the loading time of the t_{ij} variables can be found from the U_{ij} and L_{ij} variables.

5.2.2 Mathematical Model II

As mentioned at the beginning of this chapter, minimizing makespan is equivalent to minimizing the number of matching pairs of the receiving and shipping trucks for the *Case 2* model. The objective of mathematical model II is to minimize the number of matching pairs of the receiving trucks and the shipping trucks while product requirements are satisfied. With the same assumptions as in Model I, Model II integer programming model was developed as follows.

5.2.2.1 Notations

The following notations are used in Model II:

Integer Variables:

x_{ijk} = Number of units of product type k which transfer from receiving truck i to shipping truck j (for product routing),

Binary Variables:

$$v_{ij} = \begin{cases} 1, & \text{If any products transfer from receiving truck } i \text{ to shipping truck } j \text{ (or if the pair for receiving truck } i \text{ and shipping truck } j \text{ is selected)} \\ 0, & \text{Otherwise} \end{cases},$$

Data:

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k which is initially loaded in receiving truck i,

s_{jk} = Number of units of product type k which is initially needed for shipping truck j,

D = Delay time for truck change,

M = Big number.

5.2.2.2 Integer Programming Model (Model II)

The integer programming model with the objective of minimizing the number of matching pairs of the receiving truck and the shipping truck is presented below.

<u>Integer Programming Model of Model II for the *Case 2* Problem</u>

Min $$\sum_{i=1}^{R}\sum_{j=1}^{S} v_{ij}$$

Subject to

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \quad \textit{for all } i,k \qquad (5\text{-}15)$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \quad \textit{for all } j,k \qquad (5\text{-}16)$$

$$x_{ijk} \le M\, v_{ij}, \quad \textit{for all } i,j,k \qquad (5\text{-}17)$$

$$\textit{all variables} \ge 0.$$

This mathematical model has two decision variables. The first decision variable, v_{ij}, is for pairing. It shows whether receiving truck i and shipping truck j are paired or not. If the variable v_{ij} equals one, it implies that some products transfered from receiving truck i to shipping truck j. The second decision variable, x_{ijk}, represents the number of units of product type k that transfers from receiving truck i to shipping truck j. In other words, the variable x_{ijk} shows product routing.

Constraints (5-15) and *(5-16)* are exactly the same as *Constraints (5-2)* and *(5-3)*, respectively, of Model I. *Constraint (5–15)* ensures that the total number of units of product type k that transfer from receiving truck i to all shipping trucks is exactly the same as the number of units of product type k which was initially loaded in receiving truck i. Similarly, *constraint (5–16)* ensures that the total number of units of product type k that transfer from all receiving trucks to shipping truck j is exactly the same as the number of units of product type k needed for shipping truck j. *Constraint (5-17)* enforces the correct relationship between the x_{ijk} variables and the v_{ij} variables, and ensures that there is a product tranfer between a receiving truck and shipping truck if and only they are paired.

The number of decision variables for this integer programming model is $RS(1+N)$, including RS of binary variables and RSN of integer variables. The number of constraints is $N(RS+R+S)$ and includes RSN of inequality constraints and $N(R+S)$ of equality constraints.

Example 3 was applied to Model II and solved by LINDO. The problem (i.e., example 3) has four receiving trucks, three shipping trucks, and seven product types. Information about each receiving truck and shipping truck is presented in Table 14. For this example problem, there are a total of 96 variables and 133 constraints. The solution obtained for the problem using Model II is presented in Table 15. Table 15 shows the matching pairs for the receiving and shipping trucks and the product routing between them. The minimum number of matching pairs required for this example is eight as shown in Table 15.

5.2.3 Heuristic Method

For the *Case 2* problem, heuristic algorithms were developed to minimize the number of matching pairs of receiving and shipping trucks because minimizing the number of matching pairs of the receiving and shipping trucks is equivalent to minimizing the delay time due to truck changes. Minimizing delay time will minimize makespan in the *Case 2* problem.

Table 14. Example Set 3 to Illustrate Model II for the *Case 2* Problem

Receiving Trucks

Truck	**Product**	**Quantity**
1	1	150
	2	50
	3	50
2	4	200
	5	50
3	1	150
	3	150
4	6	100
	7	50

Shipping Trucks

Truck	**Product**	**Quantity**
1	1	100
	3	50
	5	50
2	2	50
	4	200
	6	100
3	1	200
	3	150
	7	50

Table 15. Number of Matching Pairs and Product Routing generated by Model II for Example 3

Receiving Truck	**Shipping Truck**	**Product Type**	**Number of Products Transferred**
1	2	2	50
1	3	1	150
		3	50
2	1	5	50
2	2	4	200
3	1	1	100
		3	50
3	3	1	50
		3	100
4	2	6	100
4	3	7	50

Six heuristic algorithms were developed and tested for the *Case 2* problem. All six heuristics are iterative algorithms. In each iteration, the best matching pair is chosen based on the selection criteria. After the best matching pair is selected, the remaining number of products in the receiving trucks and the shipping trucks are updated. The selection and the updating procedures are continued until product requirements for all receiving and shipping trucks are satisfied. The six heuristic algorithms are as follows:

1. Heuristic Algorithm 1 - Maximum flow between pairs.
2. Heuristic Algorithm 2 - Maximum ratio between pairs.
3. Heuristic Algorithm 3 - Maximum fitness between pairs.
4. Heuristic Algorithm 4 - Maximum flow with priority assignment.
5. Heuristic Algorithm 5 - Maximum ratio with priority assignment.
6. Heuristic Algorithm 6 - Maximum fitness with priority assignment.

Heuristic algorithms 1, 2 and 3 follow the same procedures except they employ different criteria for selecting the best matching pair. In each iteration, the best pair is chosen based on the selection criterion. Heuristic algorithms 4, 5 and 6 are the modified versions of heuristic algorithms 1, 2, and 3, respectively. They use the same criteria as the first three algorithms, but they also have a condition for priority assignment. In a given iteration, if there are pairs that satisfy the priority condition, heuristic algorithms 1, 2 or 3 are only applied to the pairs that satisfy the priority condition. Then, the best matching pair is chosen among those pairs that satisfy the priority condition. If there are no pairs that satisfy the priority condition in a given iteration, the process automatically reverts to heuristic algorithms 1, 2 or 3 in a given iteration. The priority condition of the heuristic algorithm for the *Case 2* problem is defined as follows:

PRIORITY CONDITION

Suppose that a certain product type is loaded in only one receiving truck during a given iteration of the heuristic algorithm. This implies that all shipping trucks that need the product type will be paired with the receiving truck that carries the product. For example, suppose that there are three receiving trucks and four shipping trucks. In a given iteration, product type 4 is only loaded in receiving truck 1. Receiving trucks 2 and 3 do not carry product type 4. Meanwhile, shipping trucks 1, 2 and 4 need to load product type 4. Shipping truck 3 does not need to load product type 4. Then, receiving truck 1 and shipping truck 1 must be paired because product type 4 is only loaded in receiving truck 1. Similarly, receiving truck 1 and shipping truck 2 have to be paired and receiving truck 1 and shipping truck 4 must also be paired.

The same argument also goes for shipping trucks. Suppose that a certain product type is needed by only one shipping truck in a given iteration. Then, all receiving trucks that carry the product must be paired with the shipping truck that needs to load the product. Therefore, if any pairs of receiving and shipping trucks satisfy one of the above two conditions in a given iteration of the heuristic algorithm, then priority is assigned to those pairs because the receiving and shipping trucks must be paired in the algorithm.

5.2.3.1 Heuristic Algorithm 1 - Maximum Flow between Pairs

For the first heuristic algorithm, the total number of products that can transfer from a receiving truck to a shipping truck is calculated for each pair of receiving and shipping trucks in a given iteration. Then, the pair that has the largest number of products transferring from a receiving truck to a shipping truck is chosen. After the best matching pair is selected, the remaining number of products in the receiving trucks and the shipping trucks are updated. The above procedures are continued until product requirements for all receiving and shipping trucks are satisfied.

HEURISTIC ALGORITHM 1

STEP 1

For each pair of receiving truck i and shipping truck j, calculate the total number of products that can transfer from the receiving truck to the shipping truck as follows:

$$\alpha_{ij} = \sum_{k=1}^{N} \min(r'_{ik}, s'_{jk}) \qquad (5\text{-}18)$$

where,

α_{ij} = Total number of products that transfer from receiving truck i to shipping truck j,

r'_{ik} = Number of units of product type k which are loaded in receiving truck i in a given iteration,

s'_{jk} = Number of units of product type k which are needed for shipping truck j in a given iteration.

STEP 2

If all α_{ij} are zero, stop; a solution for the *Case 2* problem is found. The solution is presented as the number of matching pairs of receiving and shipping trucks. If any α_{ij} is nonzero, choose the pair that has the highest α_{ij}. If there is a tie, choose a pair arbitrarily.

STEP 3

Update the remaining number of units of products in the receiving trucks and the shipping trucks. Go to *Step 1*.

5.2.3.2 Heuristic Algorithm 2 - Maximum Ratio between Pairs

For heuristic algorithm 2, the ratio of the pair, β_{ij}, is developed to select the best matching pair in a given iteration. The ratio β_{ij} for the pair of receiving truck i and shipping truck j is expressed as follows:

$$\beta_{ij} = \frac{\sum_{\substack{k=1 \\ r'_{ik}\neq 0\, or\, s'_{jk}\neq 0}}^{N} \min(r'_{ik}, s'_{jk})}{\sum_{\substack{k=1 \\ r'_{ik}\neq 0\, or\, s'_{jk}\neq 0}}^{N} \max(r'_{ik}, s'_{jk})} \quad (5\text{-}19)$$

where,

β_{ij} = Ratio for the pair of receiving truck i and shipping truck j,

r'_{ik} = Number of units of product type k which are loaded in receiving truck i in a given iteration,

s'_{jk} = Number of units of product type k which are needed for shipping truck j in a given iteration.

If the denominator equals zero, it is assumed that $\beta_{ij} = 0$.

The ratio of the pair, β_{ij}, can be considered as the correlation between receiving truck i and shipping truck j. The range of the ratio β_{ij} is between 0 and 1. Ratio $\beta_{ij} = 0$ implies that the receiving and shipping trucks have no relationship. In other words, no product in the receiving truck is needed for the shipping truck. Ratio $\beta_{ij} = 1$ implies that the number of products and the types of products for the receiving and shipping trucks are exactly the same.

HEURISTIC ALGORITHM 2

STEP 1

For each pair of receiving truck i and shipping truck j, calculate the '*ratio*' as presented in *equation (5-19)*.

STEP 2

If all β_{ij} are zero, stop; a solution for the *Case 2* problem is found. The solution is presented as the number of matching pairs of receiving and shipping trucks. If any β_{ij} is nonzero, choose the pair that has the highest β_{ij}. If there is a tie, choose the pair that has the highest α_{ij}, which can be obtained from *equation (5-18)*.

STEP 3

Update the remaining number of units of products in the receiving trucks and the shipping trucks. Go to *Step 1*.

5.2.3.3 Heuristic Algorithm 3 - Maximum Fitness between Pairs

For the third heuristic algorithm, the fitness of the pair, σ_{ij}, is developed. The fitness of the pair, σ_{ij}, can be considered as the correlation between the receiving truck and the shipping truck. However, it differs from the ratio β_{ij} in heuristic algorithm 2 because the fitness σ_{ij} gives uniform weight between product types regardless of the number of products loaded.

The range of the fitness σ_{ij} is also between 0 and 1. Fitness $\sigma_{ij} = 0$ implies that the receiving and shipping trucks have no relationship. In other words, no product in the receiving truck is needed for the shipping truck. Fitness $\sigma_{ij} = 1$ implies that the number of products and the types of products for the receiving and shipping trucks are exactly the same. The fitness σ_{ij} of the pair of receiving truck i and shipping truck j is expressed as follows.

$$\sigma_{ij} = \frac{\sum_{\substack{k=1 \\ r'_{ij} \neq 0 \, or \, s'_{jk} \neq 0}}^{N} \frac{\min(r'_{ik}, s'_{jk})}{\max(r'_{ik}, s'_{jk})}}{\sum_{\substack{k=1 \\ r'_{ij} \neq 0 \, or \, s'_{jk} \neq 0}}^{N} 1} \tag{5-20}$$

where,

σ_{ij} = Fitness of the pair of receiving truck i and shipping truck j,

r'_{ik} = Number of units of product type k which are loaded in receiving truck i in a given iteration,

s'_{jk} = Number of units of product type k which are needed for shipping truck j in a given iteration.

The denominator represents the total number of product types which is loaded in either receiving truck i or shipping truck j. If the denominator equals zero, it is assumed that $\sigma_{ij} = 0$.

HEURISTIC ALGORITHM 3

STEP 1

For each pair of receiving truck i and shipping truck j, calculate the '*fitness*' as presented in *equation (5-20)*.

STEP 2

If all σ_{ij} are zero, stop; a solution for the *Case 2* problem is found. The solution is presented as the number of matching pairs of receiving and shipping trucks. If any σ_{ij} is nonzero, choose the pair that has the highest σ_{ij}. If there is a tie, choose the pair that has the highest α_{ij} which can be obtained from *Equation (5-18)*.

STEP 3

Update the remaining number of units of products in the receiving trucks and the shipping trucks. Go to *Step 1*.

As pointed out earlier, heuristic algorithms 1, 2 and 3 follow the same procedures except that they employ different criteria for selecting the best matching pair. Figure 11 describes the algorithmic steps of heuristic algorithms 1, 2 and 3. To illustrate how they work, heuristic algorithm 3 is applied to Example 3 presented in Section 5.2.2.2. The step by step procedure is presented as follows:

STEP 1

For each pair of the receiving and shipping trucks, calculate the '*fitness*' σ_{ij} as follows:

		Shipping Truck 1	2	3
Receiving Truck	1	0.417	0.200	0.270
	2	0.250	0.250	0.000
	3	0.330	0.000	0.583
	4	0.000	0.250	0.250

Fitness σ_{ij} is calculated using *Equation (5-20)*.

For example, the fitness of receiving truck 1 and shipping truck 1, σ_{11}, is

$$\sigma_{11} = \frac{\left(\frac{100}{150}\right) + \left(\frac{50}{50}\right)}{4} = 0.417.$$

The fitness of receiving truck 3 and shipping truck 3, σ_{33}, is

$$\sigma_{33} = \frac{\left(\frac{150}{200}\right) + \left(\frac{150}{150}\right)}{3} = 0.583.$$

STEP 2

The pair of receiving truck 3 and shipping truck 3 is chosen since it has the highest fitness, $\sigma_{33} = 0.583$.

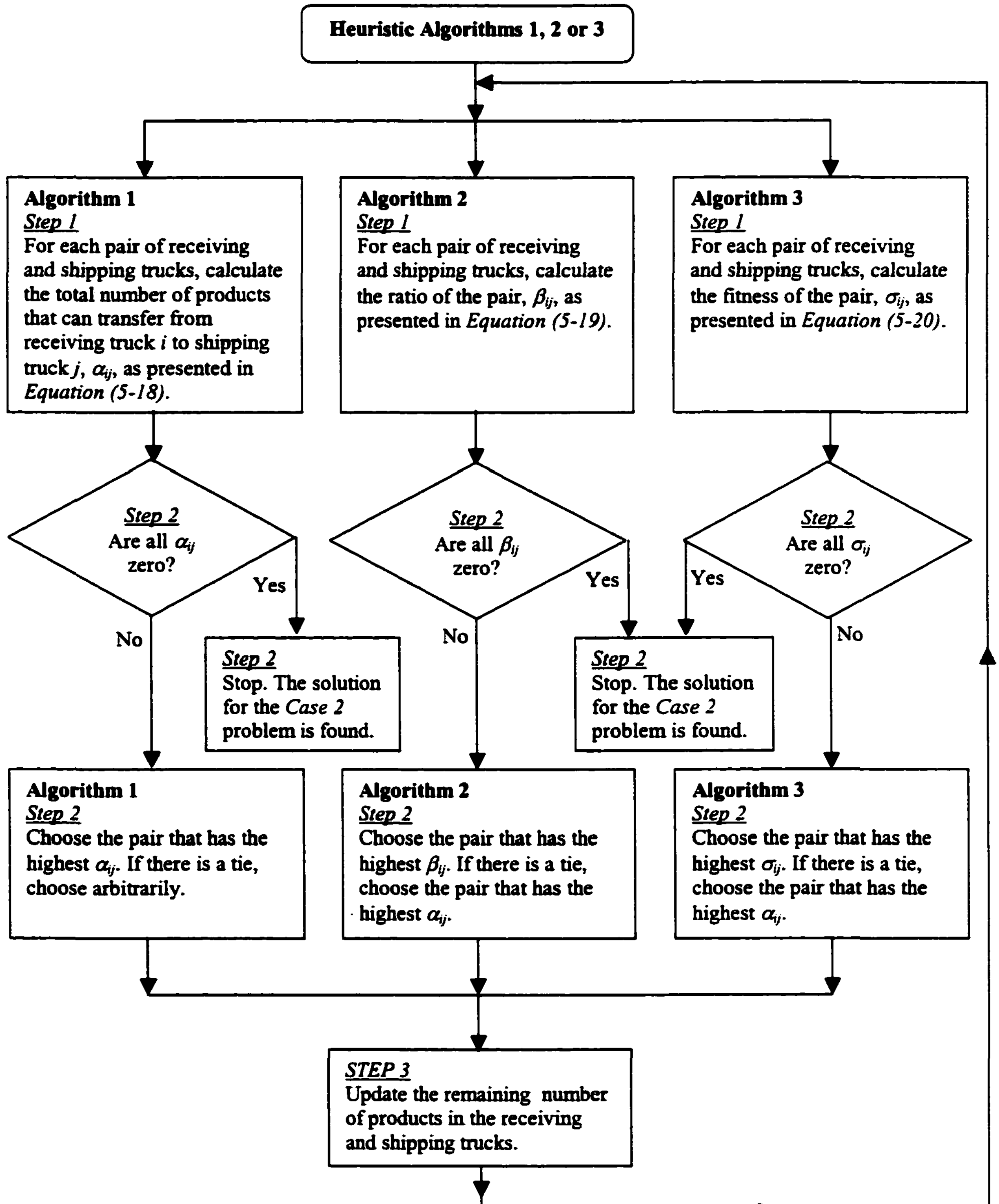


Figure 11. Heuristic Algorithms 1, 2 and 3 for the *Case 2* Problem

STEP 3

Update the remaining number of of products in the receiving trucks and the shipping trucks. To update the number of products, it only need to update the number of products in receiving truck 3 and shipping truck 3 since they are chosen in *Step 2*. The number of products in the rest of the trucks is unchanged. The updated information for the receiving and shipping trucks are as presented below:

Receiving Trucks

Truck	Product	Quantity
1	1	150
	2	50
	3	50
2	4	200
	5	50
3	1	0
	3	0
4	6	100
	7	50

Shipping Trucks

Truck	Product	Quantity
1	1	100
	3	50
	5	50
2	2	50
	4	200
	6	100
3	1	50
	3	0
	7	50

Go to *Step 1*.

STEP 1

For each pair of the receiving and shipping trucks, calculate the '*fitness*' σ_{ij} using the updated information.

Receiving Truck \ *Shipping Truck*	1	2	3
1	0.417	0.200	0.083
2	0.250	0.250	0.000
3	0.000	0.000	0.000
4	0.000	0.250	0.333

STEP 2

The pair of receiving truck 1 and shipping truck 1 is chosen since it has the largest fitness, $\sigma_{11} = 0.417$.

STEP 3

Update the remaining number of units of products in the receiving trucks and the shipping trucks. To update the number of product, it only need to update the number of products in receiving truck 1 and shipping truck 1 since they are chosen in *Step 2*. The number of products in the rest of the trucks is unchanged. The updated information for the receiving and shipping trucks are as presented below:

Receiving Trucks

Truck	Product	Quantity
1	1	50
	2	50
	3	0
2	4	200
	5	50
3	1	0
	3	0
4	6	100
	7	50

Shipping Trucks

Truck	Product	Quantity
1	1	0
	3	0
	5	50
2	2	50
	4	200
	6	100
3	1	50
	3	0
	7	50

Go to *Step 1*.

The procedure is continued until all σ_{ij} equal zero.

Heuristic algorithms 1 and 2 can be similarly applied to Example 3 without major modification. They follow the same procedure as heuristic algorithm 3 except that they use the flow factor α_{ij} or the ratio factor β_{ij} instead of the fitness factor σ_{ij}. Heuristic algorithm 3 found eight matching pairs as presented in Table 16. Even though Model II and heuristic algorithm 3 found the same number of matching pairs, the product routing is different between them.

5.2.3.4 Heuristic Algorithm 4 - Maximum Flow with Priority Assignment

Heuristic algorithm 4 is exactly the same as heuristic algorithm 1 except if there are any pairs that satisfy the priority condition in a given iteration, priority is assigned to those pairs and heuristic algorithm 1 is only applied to those pairs. Otherwise, heuristic algorithm 4 automatically reverts to heuristic algorithms 1 in a given iteration.

Table 16. Number of Matching Pairs and Product Route generated from Heuristic Algorithm 3 for Example 3

Receiving Truck	Shipping Truck	Product Type	Number of Products Transferred
1	1	1	100
		3	50
1	2	2	50
1	3	1	50
2	1	5	50
2	2	4	200
3	3	1	150
		3	150
4	2	6	100
4	3	7	50

HEURISTIC ALGORITHM 4

STEP 1

Identify all pairs that satisfy the priority condition presented at the beginning of Section 5.2.3. In other words, if a certain product type is only loaded in one receiving truck, identify all shipping trucks that need to load that particular type of product. If a certain product type is needed by only one shipping truck, identify all receiving trucks that load that particular type of product.

STEP 2

If there are any pairs that satisfy the priority condition, go to *2a* in *STEP 2*. Otherwise, go to *2b* in *STEP 2*.

2a

Apply *Heuristic Algorithm 1* to all pairs that satisfy the priority condition.

Go to *STEP 1*.

2b

Apply *Heuristic Algorithm 1*.

Go to *STEP 1*.

5.2.3.5 Heuristic Algorithm 5 - Maximum Ratio with Priority Assignment

Heuristic algorithm 5 is exactly the same as heuristic algorithm 2 except it assigns priority to the pairs that satisfy the priority condition.

HEURISTIC ALGORITHM 5

STEP 1

Identify all pairs that satisfy the priority condition presented at the beginning of Section 5.2.3.

STEP 2

If there are any pairs that satisfy the priority condition, go to *2a* in *STEP 2*. Otherwise, go to *2b* in *STEP 2*.

2a

Apply *Heuristic Algorithm 2* to all pairs that satisfy the priority condition.

Go to *STEP 1*.

2b

Apply *Heuristic Algorithm 2*.

Go to *STEP 1*.

5.2.3.6 Heuristic Algorithm 6 - Maximum Fitness with Priority Assignment

Heuristic algorithm 6 is exactly the same as heuristic algorithm 3 except it assigns priority to the pairs that satisfy the priority condition.

HEURISTIC ALGORITHM 6

STEP 1

Identify all pairs that satisfy the priority condition presented at the beginning of Section 5.2.3.

STEP 2

If there are any pairs that satisfy the priority condition, go to *2a* in *STEP 2*. Otherwise, go to *2b* in *STEP 2*.

2a

Apply *Heuristic Algorithm 3* to all pairs that satisfy the priority condition.

Go to *STEP 1*.

2b

Apply *Heuristic Algorithm 3*.

Go to *STEP 1*.

Figure 12 describes the algorithmic steps of the heuristic algorithms 4, 5 and 6. To illustrate how the modified versions of heuristic algorithms 1, 2, and 3 work, heuristic algorithm 6 is applied to Example 3 presented in Section 5.2.2.2. Heuristic algorithms 4 and 5 can be similarly applied as heuristic algorithm 6 without major modification. They follow the same procedure as heuristic algorithm 6 except that they use the flow factor α_{ij} or the ratio factor β_{ij} instead of the fitness factor σ_{ij}. The step by step procedure is presented as follows:

STEP 1

To identify the pairs that satisfy the priority condition, each product type is examined:

Product type 1: Loaded in receiving trucks 1 and 3 &
Needed for shipping trucks 1 and 3.

Product type 2: Loaded in receiving truck 1 & Needed for shipping truck 2.

Product type 3: Loaded in receiving trucks 1 and 3 &
Needed for shipping trucks 1 and 3.

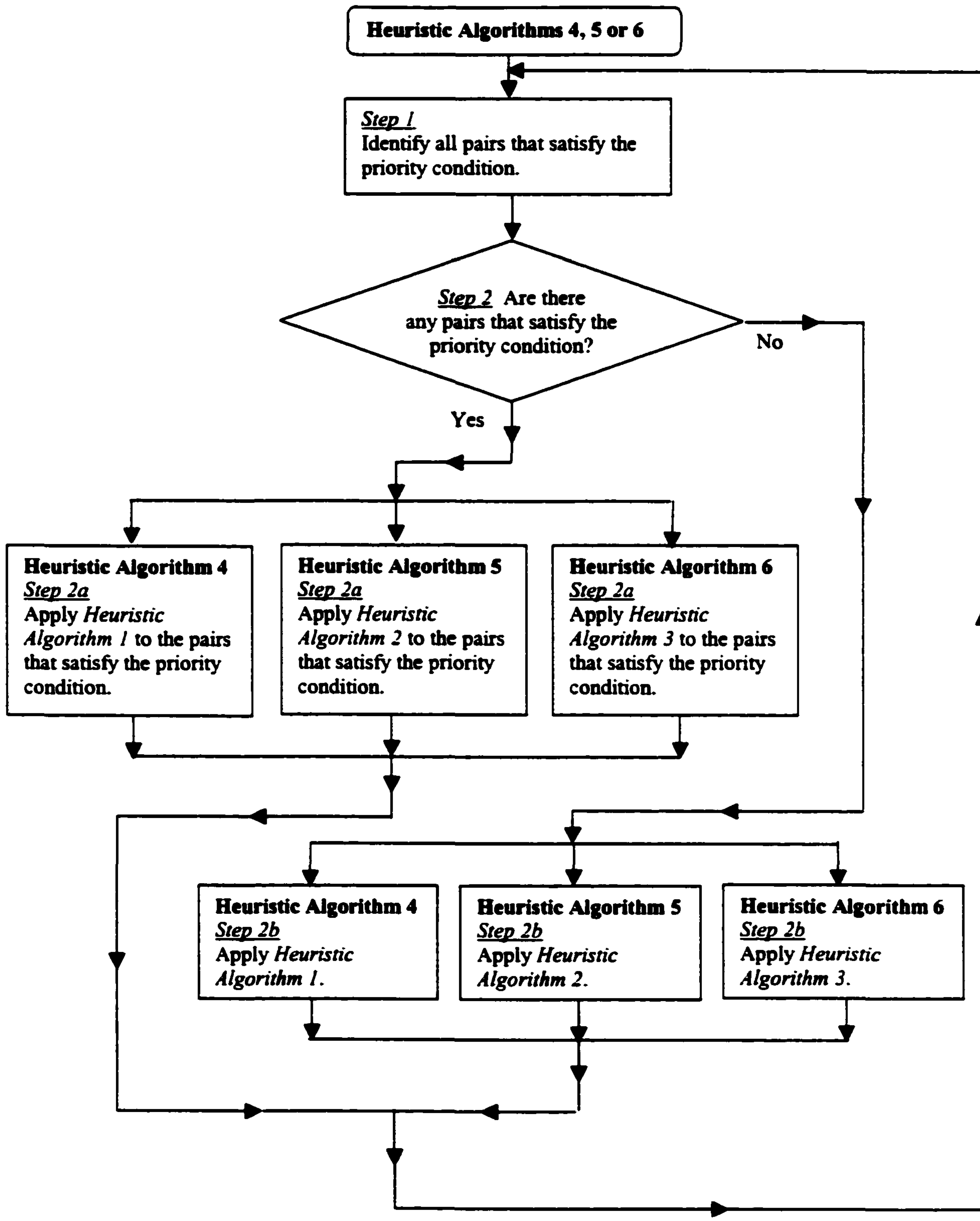


Figure 12. Heuristic Algorithms 4, 5 and 6 for the *Case 2* Problem

Product type 4: Loaded in receiving truck 2 & Needed for shipping truck 2.

Product type 5: Loaded in receiving truck 2 & Needed for shipping truck 1.

Product type 6: Loaded in receiving truck 4 & Needed for shipping truck 2.

Product type 7: Loaded in receiving truck 4 & Needed for shipping truck 3.

Therefore, the following five pairs satisfy the priority conditions:

{(Receiving Truck 1 and Shipping truck 2), (Receiving Truck 2 and Shipping truck 1), (Receiving Truck 2 and Shipping truck 2), (Receiving Truck 4 and Shipping truck 2), (Receiving Truck 4 and Shipping truck 3)}.

STEP 2

Because there are pairs that satisfy the priority condition, go to *2a* in *STEP 2.*

2a

For all pairs that satisfy the priority condition, apply *Heuristic Algorithm 3.*

STEP 1 of Heuristic Algorithm 3

Calculate the '*fitness*' σ_{ij} for all pairs that satisfy the priority condition as follow:

		Shipping Truck 1	2	3
Receiving Truck	1	×	0.200	×
	2	0.250	0.250	×
	3	×	×	×
	4	×	0.250	0.250

Fitness σ_{ij} is calculated using *Equation (5-20).*

For example, the fitness of receiving truck 1 and shipping truck 2, σ_{12}, is

$$\sigma_{12} = \frac{\left(\frac{50}{50}\right)}{5} = 0.200.$$

The fitness of receiving truck 2 and shipping truck 2, σ_{22}, is

$$\sigma_{22} = \frac{\left(\frac{200}{200}\right)}{4} = 0.250.$$

STEP 2 of Heuristic Algorithm 3

Because the fitness factors $\sigma_{21} = \sigma_{22} = \sigma_{42} = \sigma_{43} = 0.250$, the flow factors α_{21}, α_{22}, α_{42} and α_{43} need to be calculated.

$\alpha_{21} = 50$ (50 units of product type 5), $\alpha_{22} = 200$ (200 units of product type 4),

$\alpha_{42} = 100$ (100 units of product type 6), $\alpha_{43} = 50$ (50 units of product type 7).

Because the flow factor α_{22} has the highest value, choose the pair of receiving truck 2 and shipping truck 2. Note that the pair of receiving truck 3 and shipping truck 3 is chosen in the first iteration of *Heuristic Algorithm 3*. However the pair of receiving truck 3 and shipping truck 3 cannot be chosen in *Heuristic Algorithm 6* because it does not satisfy the priority condition.

STEP 3 of Heuristic Algorithm 3

Update the remaining number of products in the receiving trucks and the shipping trucks.

Receiving Trucks

Truck	Product	Quantity
1	1	150
	2	50
	3	50
2	4	0
	5	50
3	1	150
	3	150
4	6	100
	7	50

Shipping Trucks

Truck	Product	Quantity
1	1	100
	3	50
	5	50
2	2	50
	4	0
	6	100
3	1	200
	3	150
	7	50

Go to *Step 1* of *Heuristic Algorithm 6*.

STEP 1

Identify the pairs that satisfy the priority condition:

Product type 1: Loaded in receiving trucks 1 and 3 &
Needed for shipping trucks 1 and 3.

Product type 2: Loaded in receiving truck 1 & Needed for shipping truck 2.

Product type 3: Loaded in receiving trucks 1 and 3 &
Needed for shipping trucks 1 and 3.

Product type 4: No product type 4 exists anymore.

Product type 5: Loaded in receiving truck 2 & Needed for shipping truck 1.

Product type 6: Loaded in receiving truck 4 & Needed for shipping truck 2.

Product type 7: Loaded in receiving truck 4 & Needed for shipping truck 3.

Therefore, the following four pairs satisfy the priority conditions:

{(Receiving Truck 1 and Shipping truck 2), (Receiving Truck 2 and Shipping truck 1), (Receiving Truck 4 and Shipping truck 2), (Receiving Truck 4 and Shipping truck 3)}.

STEP 2

Because there are pairs that satisfy the priority condition, go to *2a* in *STEP 2*.

2a

For all pairs that satisfy the priority condition, apply *Heuristic Algorithm 3*.

STEP 1 of Heuristic Algorithm 3

Calculate the '*fitness*' σ_{ij} for all pairs that satisfy the priority condition as follow:

Receiving Truck	*Shipping Truck* 1	2	3
1	×	0.250	×
2	0.333	×	×
3	×	×	×
4	×	0.333	0.250

STEP 2 of Heuristic Algorithm 3

Because the fitness factors $\sigma_{21} = \sigma_{42} = 0.333$, the flow factors α_{21} and α_{42} need to be calculated.

$\alpha_{21} = 50$ (50 units of product type 5), $\alpha_{42} = 100$ (100 units of product type 6).

Because the flow factor α_{42} has the highest value, choose the pair of receiving truck 4 and shipping truck 2.

STEP 3 of Heuristic Algorithm 3

Update the remaining number of products in the receiving trucks and the shipping trucks.

Receiving Trucks

Truck	Product	Quantity
1	1	150
	2	50
	3	50
2	4	0
	5	50
3	1	150
	3	150
4	6	0
	7	50

Shipping Trucks

Truck	Product	Quantity
1	1	100
	3	50
	5	50
2	2	50
	4	0
	6	0
3	1	200
	3	150
	7	50

The procedure is continued until all σ_{ij} equal zero. Heuristic algorithm 6 found eight matching pairs and the solution is the same as the solution of heuristic algorithm 3 as presented in Table 16.

5.2.4 Makespan

Once the number of matching pairs of receiving and shipping trucks and product routing are know from a solution for the *Case 2* problem, makespan is calculated as follows:

$$Makespan = \sum_{i=1}^{R}\sum_{k=1}^{N} m_k r_{ik} + (P-1)D + V \qquad (5\text{-}21)$$

where,

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k which is initially loaded in receiving truck i,

s_{jk} = Number of units of product type k which is initially needed for shipping truck j,

u_k = Unloading time for one unit of product type k from a receiving truck,

l_k = Loading time for one unit of product type k to a shipping truck,

m_k = Effective time required to unload or load one unit of product type k, $m_k = \max(u_k, l_k)$,

D = Delay time for truck change,

V = Moving or travel time of products from the receiving dock to the shipping dock,

P = Number of matching pairs of receiving and shipping trucks in the solution for the *Case 2* problem.

The first term, $\sum_{i=1}^{R}\sum_{k=1}^{N} m_k r_{ik}$, represents the total required time for unloading all products from all receiving trucks and loading them to all shipping trucks in the set. Note that unloading and loading can be done simultaneously after the first V units of time have passed. Therefore, it is the larger of the two parameters u_k, and l_k that affects the makespan. The first term can be replaced with $\sum_{j=1}^{S}\sum_{k=1}^{N} m_k s_{jk}$. The second term, *(P-1)D*, presents the delay time for truck changes. Because the number of pairs in the solution is P, the total number of truck changes required is $(P-1)$. The last term, V, represents the required time for products to travel from a receiving dock to a shipping dock.

If it is assumed that loading time and unloading time are the same for all types of products and they are one unit of time, *Equation (5-21)* is simplified as follows:

$$Makespan = \sum_{i=1}^{R}\sum_{k=1}^{N} r_{ik} + (P-1)D + V . \qquad (5\text{-}22)$$

The first term, $\sum_{i=1}^{R}\sum_{k=1}^{N} r_{ik}$, presents the total number of products in the set; note that it can be replaced with $\sum_{j=1}^{S}\sum_{k=1}^{N} s_{jk}$. The second and third terms are the same as in *Equation (5-21)*.

5.2.5 Sequencing of Receiving and Shipping Trucks

The second mathematical model and the heuristic algorithms found the minimum number of matching pairs of the receiving and shipping trucks instead of finding the receiving and shipping truck sequences. However, the real interest is not the minimum number of matching pairs but the best sequences for truck spotting for both receiving and shipping trucks. Therefore, there is the need to develop a method that is able to convert the matching pairs to the best spotting sequences for the receiving and shipping trucks.

As shown in Table 15 in Section 5.2.2.2, the minimum number of matching pairs required for receiving and shipping trucks was eight for Example 3. They are as follows:

$$\{(t^r_1, t^s_2), (t^r_1, t^s_3), (t^r_2, t^s_1), (t^r_2, t^s_2), (t^r_3, t^s_1), (t^r_3, t^s_3), (t^r_4, t^s_2), (t^r_4, t^s_3)\}.$$

How can the above matching pairs be converted to near optimal sequences for the receiving and shipping trucks? It can be done by choosing any pair arbitrarily and placing the receiving truck at the end of the receiving truck sequence and placing the shipping truck at the end of the shipping truck sequence. The order of the selection of the matching pairs does not affect makespan. If the pairs are chosen from the left to the right direction, the receiving and the shipping truck sequences can be constructed as follows:

Receiving Truck Sequence : $t^r_1 \rightarrow t^r_1 \rightarrow t^r_2 \rightarrow t^r_2 \rightarrow t^r_3 \rightarrow t^r_3 \rightarrow t^r_4 \rightarrow t^r_4$.

Shipping Truck Sequence : $t^s_2 \rightarrow t^s_3 \rightarrow t^s_1 \rightarrow t^s_2 \rightarrow t^s_1 \rightarrow t^s_3 \rightarrow t^s_2 \rightarrow t^s_3$.

After removing the redundant sequences in the receiving truck sequence, the final sequences are presented as follows:

Receiving Truck Sequence : $t^r_1 \rightarrow t^r_2 \rightarrow t^r_3 \rightarrow t^r_4$.

Shipping Truck Sequence : $t^s_2 \rightarrow t^s_3 \rightarrow t^s_1 \rightarrow t^s_2 \rightarrow t^s_1 \rightarrow t^s_3 \rightarrow t^s_2 \rightarrow t^s_3$.

Now suppose the pair that has the lowest shipping truck identification number is scheduled first; then the sequences for the receiving and shipping trucks can be presented as follows:

Receiving Truck Sequence : $t^r_2 \rightarrow t^r_3 \rightarrow t^r_1 \rightarrow t^r_2 \rightarrow t^r_4 \rightarrow t^r_1 \rightarrow t^r_3 \rightarrow t^r_4$.

Shipping Truck Sequence : $t^s_1 \rightarrow t^s_1 \rightarrow t^s_2 \rightarrow t^s_2 \rightarrow t^s_2 \rightarrow t^s_3 \rightarrow t^s_3 \rightarrow t^s_3$.

After the redundant sequences in the shipping truck sequence are removed, the final sequences are presented as follows:

Receiving Truck Sequence : $t^r_2 \rightarrow t^r_3 \rightarrow t^r_1 \rightarrow t^r_2 \rightarrow t^r_4 \rightarrow t^r_1 \rightarrow t^r_3 \rightarrow t^r_4$.

Shipping Truck Sequence : $t^s_1 \rightarrow t^s_2 \rightarrow t^s_3$.

For both sequences presented above, makespan is the same regardless of the order of the pairs. Furthermore, as presented in *Equation (5-21)* or *(5-22)* in the previous section, the number of matching pairs only affects makespan, while the order of the pairs does not. Therefore, within the schedule that produces the minimum number of the matching pairs of the receiving and shipping trucks, the matching pairs can be arranged in any order to construct the schedule. For example, if there is a truck that needs to be released early, priority can be given to the pair that includes the truck and then assign the pair to the earliest available sequence positions.

5.2.5.1 Sequencing of Receiving and Shipping Trucks to minimize the Mean Flow Time for Receiving and Shipping Trucks with the given Number of Matching Pairs for the *Case 2* Problem

As mentioned above, makespan is the same regardless of the order of selection of the matching pairs for the *Case 2* problem. In this section, given the number of matching pairs as determined from the previous sections, solution approaches to minimize the mean flow time for receiving and shipping trucks in the distribution center are developed. The flow time of a receiving truck is defined as the time interval from the time the receiving truck first unloads its product or item onto the receiving dock to the time it unloads its last product or item onto the receiving dock. Similarly, the flow time of a shipping truck is defined as the time interval between the time the shipping truck first loads its needed product or item from the shipping dock to the time it loads its last product or item from the shipping dock.

The choice of optimizing on minimum flow time of trucks is justified by the fact that it is consistent with the industrial practice of minimizing the elapsed time trucks have to spend at distribution centers to deliver or pickup their items during any delivery or pickup visit to a warehouse. This way, trucks that are delivering or picking items during a given time period (e.g., a day) can be informed of their delivery or pickup times ahead of time and use the information to schedule their arrival at the warehouse to coincide or come as close as possible to the time they are scheduled for dock unloading or loading to minimize their time of stay at the warehouse site. This would allow trucks to be attended as soon as they arrive at the warehouse. The procedure is consistent with just-in-time concept. Trucks will arrive at the time they are needed and be scheduled for unloading or loading at about the same time to minimize their time of stay.

Throughout this section, it is assumed that unloading time of a product onto the receiving dock and the loading time of a product into the shipping truck are the same for all product types and it is one unit of time.

The following notations are used to develop the solution approach.

$t^r_{[i]}$ = Receiving truck that is scheduled in the i^{th} position in the matching pair sequence,

$t^s_{[i]}$ = Shipping truck that is scheduled in the i^{th} position in the matching pair sequence,

$p^t_{[i]}$ = Total number of products that transfer from receiving truck $t^r_{[i]}$ to shipping truck $t^s_{[i]}$, where ($t^r_{[i]}$, $t^s_{[i]}$) is a matching pair and scheduled in the i^{th} position in the matching pair sequence,

D = Delay time for truck change,

P = Number of matching pairs of the receiving and shipping trucks obtained from the previous solution,

b^r_i = The point in time at which receiving truck i comes the very first time into the receiving dock,

b^s_i = The point in time at which shipping truck i comes the very first time into the shipping dock,

f^r_i = The point in time at which receiving truck i leaves the warehouse after it unloads all of its products,

f^s_i = The point in time at which shipping truck i leaves the warehouse after it loads all of its needed products,

F^r_i = Flow time of receiving truck i; in other words, the amount of time receiving truck i stays in the warehouse ($= f^r_i - b^r_i$),

F^s_i = Flow time of shipping truck i; in other words, the amount of time shipping truck i stays in the warehouse ($= f^s_i - b^s_i$),

$\overline{F}$ = Mean flow time of the receiving and shipping trucks.

Then the mean flow time for the receiving and shipping trucks is defined as follows:

$$\overline{F} = \frac{\sum_{i=1}^{R} F^r_i + \sum_{j=1}^{S} F^s_j}{R+S}. \tag{5-23}$$

Suppose the matching pairs are selected and scheduled as follows:

$(t^r_{[1]}, t^s_{[1]}) \rightarrow \ldots \rightarrow (t^r_{[v]}, t^s_{[v]}) \rightarrow \ldots \rightarrow (t^r_{[w]}, t^s_{[w]}) \rightarrow \ldots \rightarrow (t^r_{[P]}, t^s_{[P]}).$ (5-24a)

then the receiving and shipping truck sequences can be presented as follows:

Receiving Truck Sequence : $t^r_{[1]} \rightarrow \ldots \rightarrow t^r_{[v]} \rightarrow \ldots \rightarrow t^r_{[w]} \rightarrow \ldots \rightarrow t^r_{[P]}.$ (5-24b)

Shipping Truck Sequence : $t^s_{[1]} \rightarrow \ldots \rightarrow t^s_{[v]} \rightarrow \ldots \rightarrow t^s_{[w]} \rightarrow \ldots \rightarrow t^s_{[P]}.$ (5-24c)

Assume receiving truck t^r_i first appears in the v^{th} position and last appears in the w^{th} position in the receiving truck sequence as presented in *(5-24b)*. Therefore, $t^r_{[v]} = t^r_i$ and $t^r_{[w]} = t^r_i$. Then, the flow time of receiving truck t^r_i, F^r_i, is calculated as follows:

$$F^r{}_i = \sum_{z=v}^{w} p^t{}_{[z]} + (w - v)D. \qquad (5\text{-}25)$$

The first term, $\sum_{z=v}^{w} p^t{}_{[z]}$, represents the time required to unload all products between the first appearance and the last appearance of receiving truck t^r_i. The second term, $(w-v)D$, represents the time required for truck changes between the first appearance and the last appearance of receiving truck t^r_i. Therefore, F^r_i presents the flow time or staying time of receiving truck t^r_i in the distribution center. Similarly, the flow time or staying time of shipping truck t^s_j, F^s_j, is calculated as in *equation (5-26)* if it is assumed shipping truck t^s_j first appears in the v^{th} position and last appears in the w^{th} position in the shipping truck sequence as presented in *(5-24c)*:

$$F^s{}_j = \sum_{z=v}^{w} p^t{}_{[z]} + (w - v)D. \qquad (5\text{-}26)$$

From this point, the receiving and shipping truck sequences will be expressed as the matching pair sequence presented in *(5-24a)* instead of expressing them individually as the receiving truck sequence as in *(5-24b)*, and the shipping truck sequence as in *(5-24c)*. Therefore, if the sequence is presented as in *(5-24a)*, then it implies there is one receiving truck sequence as presented in *(5-24b)* and one shipping truck sequence as presented in *(5-24c)*.

Consider Example 3 in Section 5.2.2.2 again. The minimum number of matching pairs required for the receiving and shipping trucks was eight as presented in Table 15 in Section 5.2.2. The pairs were as follows:

$$\{(t^r_1, t^s_2), (t^r_1, t^s_3), (t^r_2, t^s_1), (t^r_2, t^s_2), (t^r_3, t^s_1), (t^r_3, t^s_3), (t^r_4, t^s_2), (t^r_4, t^s_3)\}.$$

Assume truck change time for this example is 75 ($D = 75$).

If the pairs are chosen from the left to the right direction, the sequence is as follows:

$$(t^r_1, t^s_2) \rightarrow (t^r_1, t^s_3) \rightarrow (t^r_2, t^s_1) \rightarrow (t^r_2, t^s_2) \rightarrow (t^r_3, t^s_1) \rightarrow (t^r_3, t^s_3) \rightarrow (t^r_4, t^s_2) \rightarrow (t^r_4, t^s_3).$$

In this case, F^r_1 is calculated as follows using *equation (5-25)*:

$$F^r_1 = \sum_{z=1}^{2} p'_{[z]} + (w-v)D. = (50+200)+(2-1)75 = 325.$$

Receiving truck t^r_1 appears first in the first position and appears last in the second position in the matching pair sequence.

Similarly, F^s_3 is calculated as follows using *equation (5-26)*:

$$F^s_3 = \sum_{z=2}^{8} p'_{[z]} + (w-v)D. = (200+50+200+150+150+100+50)+(8-2)75 = 1350.$$

In this case, shipping truck t^s_3 appears first in the second position and appears last in the eighth position in the matching pair sequence.

For this sequence, the mean flow time or stay time is calculated as follows using *equation (5-23)*:

$$\overline{F} = \frac{(325+325+375+225)+(550+1350+1350)}{7} = \frac{4500}{7} = 642.86.$$

Now suppose that the pair that has the lowest shipping truck identification numbers is scheduled first; then the sequences of the receiving and shipping trucks can be presented as follows:

$$(t^r_2, t^s_1) \rightarrow (t^r_3, t^s_1) \rightarrow (t^r_1, t^s_2) \rightarrow (t^r_2, t^s_2) \rightarrow (t^r_4, t^s_2) \rightarrow (t^r_1, t^s_3) \rightarrow (t^r_3, t^s_3) \rightarrow (t^r_4, t^s_3).$$

The mean flow time for this schedule is then calculated as follows using *equation (5-23)*:

$$\overline{F} = \frac{(775+675+1225+725)+(275+500+550)}{7} = \frac{4725}{7} = 675.00.$$

As seen in the above example, the mean flow time or staying time depends on the sequences of the receiving and shipping trucks. To minimize the mean flow time of the trucks, two approaches were developed. The first approach is the enumeration method and the second method is the tabu search.

5.2.5.2 Complete Enumeration Method

From the previous solution, the number of matching pairs of the receiving and shipping trucks and product routing are already known. Therefore, the optimal solution can be found by enumerating all possible combinations for the sequences of the matching pairs. The total number of possible sequences for this problem is (*P!*) when the complete enumeration method is used.

For a small problem, it is possible to find an optimal solution with this method. However, this method is not practical for medium to large size problems. For example, suppose that the number of matching pairs is fifteen. Then the total number of possible sequences is (*15!*) = 1.3×10^{12}. In this case, it is not practical to solve the problem by enumerating all possible sequences. Therefore, what is required is a method that finds solutions within reasonable amounts of time. The next section presents the tabu search to solve the problem of minimizing the mean flow time within reasonable time duration.

The reason why the complete enumeration method is adopted in this research is to provide a basis to benchmark the performance of the tabu search. For small-sized problems, the first method is able to find the worst solution and the average solution of all possible sequences as well as the optimal solution because it enumerates over all possible sequences. Solutions obtained from the tabu search can be compared with the solutions obtained by the complete enumeration method to test the performance of the tabu search.

5.2.5.3 Tabu Search Method

The tabu search was introduced in Chapter 4 to solve *Case 1* problem. The tabu search developed for the *Case 2* problem is very similar to the tabu search used for the *Case 1* problem. The tabu search for the *Case 2* problem used the following basic elements of the neighborhood search procedure:

1. *Initial Sequence* - The initial sequence is randomly picked for the *Case 2* problem.
2. *Neighborhood of the Current Solution* - The adjacent pairwise interchange operation is used to generate a neighborhood of a current solution. Suppose the current receiving truck sequence and shipping truck sequence are scheduled as follows:

$$(t^r_{[1]}, t^s_{[1]}) \rightarrow (t^r_{[2]}, t^s_{[2]}) \rightarrow (t^r_{[3]}, t^s_{[3]}) \rightarrow \ldots \rightarrow (t^r_{[P-1]}, t^s_{[P-1]}) \rightarrow (t^r_{[P]}, t^s_{[P]}).$$

Then the neighborhood of the current solution would be exactly the following $(P-1)$ sequences:

$$(t^r_{[2]}, t^s_{[2]}) \rightarrow (t^r_{[1]}, t^s_{[1]}) \rightarrow (t^r_{[3]}, t^s_{[3]}) \rightarrow \ldots \rightarrow (t^r_{[P-1]}, t^s_{[P-1]}) \rightarrow (t^r_{[P]}, t^s_{[P]}).$$

(Interchange the matching pairs $(t^r_{[1]}, t^s_{[1]})$ and $(t^r_{[2]}, t^s_{[2]})$).

$$(t^r_{[1]}, t^s_{[1]}) \rightarrow (t^r_{[3]}, t^s_{[3]}) \rightarrow (t^r_{[2]}, t^s_{[2]}) \rightarrow \ldots \rightarrow (t^r_{[P-1]}, t^s_{[P-1]}) \rightarrow (t^r_{[P]}, t^s_{[P]}).$$

(Interchange the matching pairs $(t^r_{[2]}, t^s_{[2]})$ and $(t^r_{[3]}, t^s_{[3]})$).

o

o

$$(t^r_{[1]}, t^s_{[1]}) \rightarrow (t^r_{[2]}, t^s_{[2]}) \rightarrow (t^r_{[3]}, t^s_{[3]}) \rightarrow \ldots \rightarrow (t^r_{[P]}, t^s_{[P]}) \rightarrow (t^r_{[P-1]}, t^s_{[P-1]}).$$

(Interchange the matching pairs $(t^r_{[P-1]}, t^s_{[P-1]})$ and $(t^r_{[P]}, t^s_{[P]})$).

Even though this neighborhood limits the number of new choices to evaluate, it is relatively small and easy to generate. It is a trade-off against using a more complicated method of neighborhood generation that would require high computational time.

3. *Selection Criterion* - To select the next solution after the adjacent pairwise interchange, all solutions in the neighborhood are evaluated and the best solution among all neighborhood solutions is chosen as the next solution even if this makes the objective function value somewhat worse.

4. *Termination* - The tabu search will stop if there is no improvement in the objective function value after the search has been carried out with a new starting random solution or seed for a maximum number of times specified by the user. In other words, if a certain number of consecutive initial sequences generated randomly do not improve the current best solution, the algorithm will stop. For the *Case 2* problem, 100 was used as the maximum number of random initial sequences.

5. *Number of Tabu List* - Seven tabu lists were used for this algorithm.

The tabu search algorithm developed for the *Case 2* problem is as described below.

TABU SEARCH ALGORITHM FOR THE *CASE 2* PROBLEM

STEP 1

Generate the initial sequence randomly. Set the current sequence as the initial sequence. Set the iteration number equal to one and initialize the tabu list.

STEP 2

For each neighborhood sequence of the current sequence,

(2a) If the neighborhood sequence already exists in the tabu list, ignore the neighborhood sequence. Otherwise, go to *Step (2b)*.

(2b) Calculate the mean flow time, $\overline{F}$, of the receiving and shipping truck sequences for the neighborhood sequence of the current sequence.

STEP 3

Choose the next sequence as the neighborhood sequence that has the smallest $\overline{F}$. Set the current sequence as the next sequence.

STEP 4

If the current sequence is the best solution found so far, set the best sequence as the current sequence and set the iteration number and the random initial sequence number to 1. Otherwise, increase the iteration number by 1.

STEP 5

If the iteration number is greater than the maximum number of iterations allowed, increase the random initial sequence number by 1. Otherwise, go to *Step 2*.

STEP 6

If the random initial sequence number is greater than the maximum number of random initial sequences allowed, choose the best sequence as the best solution found so far and stop. Otherwise, go to *Step 1*.

This tabu search algorithm is very similar to the modified tabu search algorithm presented in Chapter 4. To enhance the quality of the solution, several initial seed sequences were used for the algorithm. The main drawback of this algorithm is that the selection of the next sequence is relatively expensive because the computational time required to evaluate each adjacent neighborhood interchange is very large. Therefore, the net change of the flow time or staying time for each adjacent neighborhood is evaluated to enhance the speed of the algorithm instead of actually calculating the mean flow time for each adjacent neighborhood. The following section presents how the net change of the flow time can be calculated.

5.2.5.4 Calculation of the Net Change of Flow Time of Adjacent Neighborhood for the Tabu Search Method for the *Case 2* Problem

Suppose the sequence is scheduled as follows and the two adjacent neighborhoods $(t^r_{[v]}, t^s_{[v]})$ and $(t^r_{[w]}, t^s_{[w]})$ are to be interchanged.

$$(t^r_{[1]}, t^s_{[1]}) \rightarrow (t^r_{[2]}, t^s_{[2]}) \rightarrow \ldots \rightarrow (t^r_{[v]}, t^s_{[v]}) \rightarrow (t^r_{[w]}, t^s_{[w]}) \rightarrow \ldots \rightarrow (t^r_{[P]}, t^s_{[P]}).$$

Interchange

After the adjacent neighborhood interchange, the sequence will become as follows:

$$(t^r_{[1]}, t^s_{[1]}) \rightarrow (t^r_{[2]}, t^s_{[2]}) \rightarrow \ldots \rightarrow (t^r_{[w]}, t^s_{[w]}) \rightarrow (t^r_{[v]}, t^s_{[v]}) \rightarrow \ldots \rightarrow (t^r_{[P]}, t^s_{[P]}).$$

As mentioned earlier, $p^t_{[i]}$ is defined as the total number of products that transfer from receiving truck $t^r_{[i]}$ to shipping truck $t^s_{[i]}$, where $(t^r_{[i]}, t^s_{[i]})$ is a matching pair and scheduled in the i^{th} position in the matching pair sequence, and D is defined as the delay time for truck change. Let $t^r_{[v]} = t^r_i$ and $t^r_{[w]} = t^r_j$. In this case, the net change of the flow time for the receiving truck sequence is calculated according to the following algorithm or steps:

<u>NET CHANGE OF FLOW TIME FOR RECEIVING TRUCK INTERCHANGE</u>

If $t^r_i = t^r_j$, $\therefore$ Net Change = 0.

Otherwise (i.e. if $t^r_i \neq t^r_j$), the net change is calculated as follows after evaluating the net change of F^r_i and F^r_j before and after the interchange.

1. If t^r_i appears only once in the receiving truck sequence and ...

   i) If t^r_j appears only once in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

   Before : $F^r_i = p^t_{[v]}$ & $F^r_j = p^t_{[w]}$

   After : $F^r_i = p^t_{[v]}$ & $F^r_j = p^t_{[w]}$

   $\therefore$ Net Change = 0.

   ii) If $t^r_{[w]}$ is the first position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

   Before : $F^r_i = p^t_{[v]}$ & $F^r_j = p^t_{[w]} + \ldots$

   After : $F^r_i = p^t_{[v]}$ & $F^r_j = p^t_{[w]} + D + p^t_{[v]} + \ldots$

   $\therefore$ Net Change = $(p^t_{[v]} + D)$.

   iii) If $t^r_{[w]}$ is a position between the first position and the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

   Before : $F^r_i = p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[v]} + D + p^t_{[w]} + \ldots$

   After : $F^r_i = p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[w]} + D + p^t_{[v]} + \ldots$

   $\therefore$ Net Change = 0.

   iv) If $t^r_{[w]}$ is the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

   Before : $F^r_i = p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[v]} + D + p^t_{[w]}$

   After : $F^r_i = p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[w]}$

   $\therefore$ Net Change = $-(p^t_{[v]} + D)$

2. If $t^r_{[v]}$ is the first position of receiving truck t^r_i in the receiving truck sequence and ...

   i) If t^r_j appears only once in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

   Before : $F^r_i = p^t_{[v]} + D + p^t_{[w]} + \ldots$ & $F^r_j = p^t_{[w]}$

   After : $F^r_i = p^t_{[v]} + \ldots$ & $F^r_j = p^t_{[w]}$

   $\therefore$ Net Change = $-(p^t_{[w]} + D)$.

ii) If $t^r_{[w]}$ is the first position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = p^t_{[v]} + D + p^t_{[w]} + \dots$ & $F^r_j = p^t_{[w]} + \dots$

After : $F^r_i = p^t_{[v]} + \dots$ & $F^r_j = p^t_{[w]} + D + p^t_{[v]} + \dots$

$\therefore$ Net Change = $(p^t_{[v]} - p^t_{[w]})$.

iii) If $t^r_{[w]}$ is the position between the first position and the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = p^t_{[v]} + D + p^t_{[w]} + \dots$ & $F^r_j = \dots + p^t_{[v]} + D + p^t_{[w]} + \dots$

After : $F^r_i = p^t_{[v]} + \dots$ & $F^r_j = \dots + p^t_{[w]} + \mathrm{D} + p^t_{[v]} + \dots$

$\therefore$ Net Change = $-(p^t_{[w]} + D)$.

iv) If $t^r_{[w]}$ is the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = p^t_{[v]} + D + p^t_{[w]} + \dots$ & $F^r_j = \dots + p^t_{[v]} + D + p^t_{[w]}$

After : $F^r_i = p^t_{[v]} + \dots$ & $F^r_j = \dots + p^t_{[w]}$

$\therefore$ Net Change = $-(p^t_{[v]} + p^t_{[w]} + 2D)$.

3. If $t^r_{[v]}$ is the position between the first position and the last position of receiving truck t^r_i in the receiving truck sequence and …

i) If t^r_j appears only once in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \dots + p^t_{[v]} + D + p^t_{[w]} + \dots$ & $F^r_j = p^t_{[w]}$

After : $F^r_i = \dots + p^t_{[w]} + D + p^t_{[v]} + \dots$ & $F^r_j = p^t_{[w]}$

$\therefore$ Net Change = 0.

ii) If $t^r_{[w]}$ is the first position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \dots + p^t_{[v]} + D + p^t_{[w]} + \dots$ & $F^r_j = p^t_{[w]} + \dots$

After : $F^r_i = \dots + p^t_{[w]} + D + p^t_{[v]} + \dots$ & $F^r_j = p^t_{[w]} + D + p^t_{[v]} + \dots$

$\therefore$ Net Change = $(p^t_{[v]} + D)$.

iii) If $t^r_{[w]}$ is the position between the first position and the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \ldots + p^t_{[v]} + D + p^t_{[w]} + \ldots$ & $F^r_j = \ldots + p^t_{[v]} + D + p^t_{[w]} + \ldots$

After : $F^r_i = \ldots + p^t_{[w]} + D + p^t_{[v]} + \ldots$ & $F^r_j = \ldots + p^t_{[w]} + D + p^t_{[v]} + \ldots$

$\therefore$ Net Change = 0.

iv) If $t^r_{[w]}$ is the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \ldots + p^t_{[v]} + D + p^t_{[w]} + \ldots$ & $F^r_j = \ldots + p^t_{[v]} + D + p^t_{[w]}$

After : $F^r_i = \ldots + p^t_{[w]} + D + p^t_{[v]} + \ldots$ & $F^r_j = \ldots + p^t_{[w]}$

$\therefore$ Net Change = $-(p^t_{[v]} + D)$.

4. If $t^r_{[v]}$ is the last position of receiving truck t^r_i in the receiving truck sequence and …

i) If t^r_j appears only once in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \ldots + p^t_{[v]}$ & $F^r_j = p^t_{[w]}$

After : $F^r_i = \ldots + p^t_{[w]} + D + p^t_{[v]}$ & $F^r_j = p^t_{[w]}$

$\therefore$ Net Change = $(p^t_{[w]} + D)$.

ii) If $t^r_{[w]}$ is the first position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \ldots + p^t_{[v]}$ & $F^r_j = p^t_{[w]} + \ldots$

After : $F^r_i = \ldots + p^t_{[w]} + D + p^t_{[v]}$ & $F^r_j = p^t_{[w]} + D + p^t_{[v]} + \ldots$

$\therefore$ Net Change = $(p^t_{[v]} + p^t_{[w]} + 2D)$.

iii) If $t^r_{[w]}$ is the position between the first position and the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \ldots + p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[v]} + D + p^t_{[w]} + \ldots$

After : $F^r_i = \ldots + p^t_{[w]} + D + p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[w]} + D + p^t_{[v]} + \ldots$

$\therefore$ Net Change = $(p^t_{[w]} + D)$.

iv) If $t^r_{[w]}$ is the last position of receiving truck t^r_j in the receiving truck sequence, then F^r_i and F^r_j before and after the interchange are as follows:

Before : $F^r_i = \ldots + p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[v]} + D + p^t_{[w]}$

After : $F^r_i = \ldots + p^t_{[w]} + D + p^t_{[v]}$ & $F^r_j = \ldots + p^t_{[w]}$

$\therefore$ Net Change = $(p^t_{[w]} - p^t_{[v]})$.

The exactly same argument goes to the net change of the flow time for the shipping truck sequence. Figure 13 presents the summary of net change of the flow time for the receiving truck sequence and Figure 14 presents the summary of net change of the flow time for the shipping truck sequence. After the net change of the flow time for the receiving truck change and shipping truck change are calculated and their sum is greater than zero, the interchange will increase the mean flow time. If their sum is zero, the mean flow time is the same before and after the interchange. If their sum is less than zero, it decreases the mean flow time after the interchange. Therefore, the neighborhood that has the smallest net change is chosen for the tabu search and it is set as the next sequence.

To explain the net change of the mean flow time, consider Example 3 in Section 5.2.2.2 again. Assume the current sequence is as presented in *(5-27)*:

$$(t^r_1, t^s_2) \rightarrow (t^r_1, t^s_3) \rightarrow (t^r_2, t^s_1) \rightarrow (t^r_2, t^s_2) \rightarrow (t^r_3, t^s_1) \rightarrow (t^r_3, t^s_3) \rightarrow (t^r_4, t^s_2) \rightarrow (t^r_4, t^s_3). \quad (5\text{-}27)$$

From Section 5.2.5.1, the mean flow time, $\overline{F}=642.86$, was found. Next, suppose the matching pairs (t^r_3, t^s_3) and (t^r_4, t^s_2) are interchanged from sequence *(5-27)*. Then the adjacent neighborhood of the current solution corresponding to this interchange is as presented in *(5-28)*:

$$(t^r_1, t^s_2) \rightarrow (t^r_1, t^s_3) \rightarrow (t^r_2, t^s_1) \rightarrow (t^r_2, t^s_2) \rightarrow (t^r_3, t^s_1) \rightarrow (t^r_4, t^s_2) \rightarrow (t^r_3, t^s_3) \rightarrow (t^r_4, t^s_3). \quad (5\text{-}28)$$

One way of finding the mean flow time, $\overline{F}$, is to calculate it using *equation (5-23)* presented in Section 5.2.5.1.

$$\overline{F} = \frac{(325+325+550+450)+(550+1125+1350)}{7} = \frac{4675}{7} = 667.86\,.$$

Let $r^r_{[v]} = r^r_i$ and $r^r_{[w]} = r^r_j$.

If $r^r_i = r^r_j$, $\therefore$ Net Change = 0.

Otherwise(i.e. if $r^r_i \neq r^r_j$), the net change is calculated as follows after evaluating the net change of F^r_i and F^r_j before and after the interchange.

1. If r^r_i appears only once in the receiving truck sequence and ...
   - i) If r^r_j appears only once in the receiving truck sequence. $\therefore$ Net Change = 0.
   - ii) If $r^r_{[w]}$ is the first position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $(p^t_{[v]} + D)$.
   - iii) If $r^r_{[w]}$ is the position between the first position and the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = 0.
   - iv) If $r^r_{[w]}$ is the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $-(p^t_{[v]} + D)$.

2. If $r^r_{[v]}$ is the first position of receiving truck r^r_i in the receiving truck sequence and ...
   - i) If r^r_j appears only once in the receiving truck sequence. $\therefore$ Net Change = $-(p^t_{[w]} + D)$.
   - ii) If $r^r_{[w]}$ is the first position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $(p^t_{[v]} - p^t_{[w]})$.
   - iii) If $r^r_{[w]}$ is the position between the first position and the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $-(p^t_{[w]} + D)$.
   - iv) If $r^r_{[w]}$ is the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $-(p^t_{[v]} + p^t_{[w]} + 2D)$.

3. If $r^r_{[v]}$ is the position between the first position and the last position of receiving truck r^r_i in the receiving truck sequence and ...
   - i) If r^r_j appears only once in the receiving truck sequence. $\therefore$ Net Change = 0.
   - ii) If $r^r_{[w]}$ is the first position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $(p^t_{[v]} + D)$.
   - iii) If $r^r_{[w]}$ is the position between the first position and the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = 0.
   - iv) If $r^r_{[w]}$ is the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $-(p^t_{[v]} + D)$.

4. If $r^r_{[v]}$ is the last position of receiving truck r^r_i in the receiving truck sequence and ...
   - i) If r^r_j appears only once in the receiving truck sequence. $\therefore$ Net Change = $(p^t_{[w]} + D)$.
   - ii) If $r^r_{[w]}$ is the first position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $(p^t_{[v]} + p^t_{[w]} + 2D)$.
   - iii) If $r^r_{[w]}$ is the position between the first position and the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $(p^t_{[w]} + D)$.
   - iv) If $r^r_{[w]}$ is the last position of receiving truck r^r_j in the receiving truck sequence. $\therefore$ Net Change = $(p^t_{[w]} - p^t_{[v]})$.

Figure 13. Net Change of the Flow Time for the Receiving Truck Interchange

Let $t^s_{[v]} = t^s_i$ and $t^s_{[w]} = t^s_j$.

If $t^s_i = t^s_j$, $\therefore$ Net Change = 0.

Otherwise(i.e. if $t^s_i \neq t^s_j$), the net change is calculated as follows after evaluating the net change of F^s_i and F^s_j before and after the interchange.

1. If t^s_i appears only once in the shipping truck sequence and ...
   i) If t^s_j appears only once in the shipping truck sequence. $\therefore$ Net Change = 0.
   ii) If $t^s_{[w]}$ is the first position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= (p^t_{[v]} + D)$.
   iii) If $t^s_{[w]}$ is the position between the first position and the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change = 0.
   iv) If $t^s_{[w]}$ is the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= -(p^t_{[v]} + D)$.

2. If $t^s_{[v]}$ is the first position of shipping truck t^s_i in the shipping truck sequence and ...
   i) If t^s_j appears only once in the shipping truck sequence. $\therefore$ Net Change $= -(p^t_{[w]} + D)$.
   ii) If $t^s_{[w]}$ is the first position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= (p^t_{[v]} - p^t_{[w]})$.
   iii) If $t^s_{[w]}$ is the position between the first position and the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= -(p^t_{[w]} + D)$.
   iv) If $t^s_{[w]}$ is the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= -(p^t_{[v]} + p^t_{[w]} + 2D)$.

3. If $t^s_{[v]}$ is the position between the first position and the last position of shipping truck t^s_i in the shipping truck sequence and ...
   i) If t^s_j appears only once in the shipping truck sequence. $\therefore$ Net Change = 0.
   ii) If $t^s_{[w]}$ is the first position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= (p^t_{[v]} + D)$.
   iii) If $t^s_{[w]}$ is the position between the first position and the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change = 0.
   iv) If $t^s_{[w]}$ is the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= -(p^t_{[v]} + D)$.

4. If $t^s_{[v]}$ is the last position of shipping truck t^s_i in the shipping truck sequence and ...
   i) If t^s_j appears only once in the shipping truck sequence. $\therefore$ Net Change $= (p^t_{[w]} + D)$.
   ii) If $t^s_{[w]}$ is the first position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= (p^t_{[v]} + p^t_{[w]} + 2D)$.
   iii) If $t^s_{[w]}$ is the position between the first position and the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= (p^t_{[w]} + D)$.
   iv) If $t^s_{[w]}$ is the last position of shipping truck t^s_j in the shipping truck sequence. $\therefore$ Net Change $= (p^t_{[w]} - p^t_{[v]})$.

Figure 14. Net Change of the Flow Time for the Shipping Truck Interchange

However, a simpler way of finding $\bar{F}$ is to use the net change equations of the flow time. Consider the matching pair sequence before the pairs (t^r_3, t^s_3) and (t^r_4, t^s_2) are interchanged as presented in sequence *(5-27)*. As can be seen, receiving trucks $t^r_{[6]} = t^r_3$ and $t^r_{[7]} = t^r_4$ in sequence *(5-27)*. Receiving truck $t^r_{[6]}$ is in the last position or appearance of receiving truck t^r_3 in the receiving truck sequence and receiving truck $t^r_{[7]}$ is the first position or appearance of receiving truck t^r_4 in the receiving truck sequence. Therefore, the net change of the flow time for the receiving truck interchange is calculated as follows:

Net Change for Receiving Truck Interchange

$$= (p^t_{[v]} + p^t_{[w]} + 2D) = (150 + 100 + 2(75)) = 400.$$

For the shipping trucks, $t^s_{[6]} = t^s_3$ and $t^s_{[7]} = t^s_2$ in sequence *(5-27)*. Shipping truck $t^s_{[6]}$ is in a position between the first appearance and the last appearance of shipping truck t^s_3 in the shipping truck sequence. Note that the first appearance in the shipping truck sequence for shipping truck t^s_3 is $t^s_{[2]}$ and the last appearance in the same sequence for shipping truck t^s_3 is $t^s_{[8]}$ in sequence *(5-27)*. Meanwhile, shipping truck $t^s_{[7]}$ is in the last appearance for shipping truck t^s_2 in sequence *(5-27)*. Therefore, the net change of the flow time for the shipping truck interchange is calculated as follows:

Net Change for Shipping Truck Interchange

$$= -(p^t_{[v]} + D). = -(150 + 75) = -225.$$

Total net change of the receiving and shipping truck interchanges will be 175 (= 400 – 225). Therefore, it will increase the mean flow time by $\frac{175}{7} = 25$. The mean flow time of the adjacent interchange as presented in sequence *(5-28)* will be 642.86 plus 25 which equals 667.86. This mean flow time, 667.86, is the same as the mean flow time found using *equation (5-23)* presented in Section 5.2.5.1.

Consider another adjacent neighborhood interchange. If the two matching pairs (t^r_1, t^s_2) and (t^r_1, t^s_3) are interchanged from sequence *(5-27)*, the new sequence will be as follows:

$$(t^r_1, t^s_3) \rightarrow (t^r_1, t^s_2) \rightarrow (t^r_2, t^s_1) \rightarrow (t^r_2, t^s_2) \rightarrow (t^r_3, t^s_1) \rightarrow (t^r_3, t^s_3) \rightarrow (t^r_4, t^s_2) \rightarrow (t^r_4, t^s_3). \quad (5\text{-}29)$$

Using *equation (5-23)* presented in Section 5.2.5.1, the mean flow time for sequence *(5-29)*, $\bar{F} = 621.43$, was found.

In order to find the mean flow time of sequence *(5-29)*, suppose the net change of flow time is used. Then, for the receiving truck interchange, it can be seen from sequence *(5-27)*, $t^r_{[1]} = t^r_1$ and $t^r_{[2]} = t^r_1$. Because receiving trucks $t^r_{[1]}$ and $t^r_{[2]}$ are the same receiving truck t^r_1 (i.e. $t^r_{[1]}$ (=t^r_1) = $t^r_{[2]}$ (=t^r_1)), the net change of flow time for receiving truck interchange is zero.

Net Change for Receiving Truck Interchange = 0.

For the shipping truck interchange, $t^s_{[1]} = t^s_2$ and $t^s_{[2]} = t^s_3$ from sequence *(5-27)*. Shipping truck $t^s_{[1]}$ is in the first appearance of shipping truck t^s_2 and shipping truck $t^s_{[2]}$ is also in the first appearance of shipping truck t^s_3 in sequence *(5-27)*. Therefore, the net change of the flow time for the shipping truck interchange is calculated as follows:

Net Change for Shipping Truck Interchange

$$= (p^t_{[v]} - p^t_{[w]}) = (50 - 200) = -150.$$

Total net change of the receiving and shipping truck interchanges will be −150 (= 0 − 150). Therefore, it will change the mean flow time by $\frac{-150}{7} = -21.43$. The mean flow time of the adjacent interchange as presented in sequence *(5-29)* will be 642.86 minus 21.43 which is 621.43. This mean flow time, 621.43, is the same as the mean flow time found using *equation (5-23)* presented in Section 5.2.5.1.

Using the net change to find the mean flow time for adjacent neighborhoods, the run time of the tabu search was drastically reduced. In the real implementation of the tabu search, the net change of the flow time was used to evaluate the adjacent neighborhood and to select the next sequence instead of using *equation (5-23)* presented in Section 5.2.5.1.

5.3 Implementation and Results

After applying mathematical programming model of Model II and the six heuristic algorithms to the same twenty sets of problems as in the *Case 1* problem, the results shown in Tables 17 and 18 were obtained. Table 17 shows the optimal solutions that were obtained using the Model II mathematical model. The solutions are presented as the number of matching pairs of the receiving and shipping trucks. The product routings for the optimal solutions are presented in Appendix C.

Table 17. Minimum Number of Matching Pairs obtained from the Mathematical Model (Model II) for the *Case 2* Problem

Problem Number	Number of Receiving Trucks	Number of Shipping Trucks	Number of Product Types	Total Possible Matching Pairs	Upper Bound of Matching Pairs	Mathematical Model II		
						Number of Variables	Number of Constraints	Optimal Solution (Matching Pairs)
1	4	5	4	20	19	100	116	11
2	5	4	6	20	19	140	174	11
3	3	3	8	9	9	81	120	8
4	5	5	8	25	21	225	280	16
5	5	3	8	15	14	135	184	11
6	4	4	5	16	14	96	120	11
7	5	4	6	20	17	140	174	11
8	3	5	7	15	13	120	161	12
9	4	4	8	16	15	144	192	12
10	3	4	9	12	11	120	171	10
11	5	4	6	20	18	140	174	11
12	6	4	8	24	20	216	272	15
13	5	6	8	30	23	270	328	17
14	5	5	8	25	25	225	280	15
15	6	5	4	30	29	150	164	13
16	5	6	6	30	26	210	246	16
17	4	4	7	16	12	128	168	11
18	6	6	7	36	26	288	336	16
19	5	5	10	25	22	275	350	16
20	6	6	9	36	30	360	432	18

Table 18. Number of Matching Pairs obtained from Heuristic Solutions for the *Case 2* Problem

Problem Number	Upper Bound of Matching Pairs	Optimal Solution (Matching Pairs)	Heuristic Solutions of the *Case 2* Problem (Matching Pairs)						
			Heuristic 1	Heuristic 2	Heuristic 3	Heuristic 4	Heuristic 5	Heuristic 6	Compound Heuristic
1	19	11	12	12	12	12	12	12	12
2	19	11	13	13	13	13	14	14	13
3	9	8	9	8	9	9	9	9	8
4	21	16	19	20	18	20	20	20	18
5	14	11	12	12	12	11	11	11	11
6	14	11	12	11	11	12	11	11	11
7	17	11	12	12	12	13	13	13	12
8	13	12	12	12	12	12	12	12	12
9	15	12	12	12	12	12	12	13	12
10	11	10	11	11	11	11	11	11	11
11	18	11	12	12	12	12	12	12	12
12	20	15	18	18	17	17	17	17	17
13	23	17	19	19	18	19	19	18	18
14	25	15	20	20	18	20	18	18	18
15	29	13	18	17	17	18	17	17	17
16	26	16	17	18	17	17	16	17	16
17	12	11	12	11	12	12	12	12	11
18	26	16	18	18	17	17	17	17	17
19	22	16	18	18	18	19	19	19	18
20	30	18	22	21	22	22	21	21	21

The results obtained from the six heuristic algorithms and the compound heuristic algorithm are presented in Table 18. Similar to the definiation in Section 4.3, the compound heuristic solution for the *Case 2* problem is the best solution found after applying all six heuristic algorithms. As can be seen in Table 18, the compound heuristic algorithm found solutions that were close to the optimal solutions. The compound heuristic algorithm found the optimal solution in only seven of the twenty test problems. However, as it can be seen, most of the compound heuristic solutions were very close to the optimal solutions. The differences between the optimal solutions and the compound heuristic solutions were one or two matching pairs in most cases. In the worst case, the difference was four in *Test Set* 15. The optimal solution was 13 and the compound heuristic solution found 17. However, the heuristic solution found a solution that was far below the upper bound of 29 for test problem set 15.

Among the six heuristic algorithms, heuristic algorithm 3 (Maximum fitness) performed the best. Heuristic algorithm 3 found the best solutions among the six heuristic solutions in fifteen out of twenty test problems. Meanwhile, the worst algorithm among the six heuristic algorithms was heuristic algorithm 1. Heuristic algorithm 1 found the best solutions among the six heuristic solutions only in eight out of twenty test problems. Moreover, heuristic algorithm 1 was dominated by heuristic algorithm 3. It means that the solutions found by heuristic algorithm 1 was worse than or equal to the solutions found by heuristic algorithm 3 in all the twenty test sets. One interesting characteristic found from Table 18 is that only heuristic algorithm 2 found the optimal solutions in two problem sets (sets 3 and 17) among the six heuristic algorithms. Similarly, only heuristic algorithm 5 found the optimal solution in test problem 16 among the six heuristic algorithms. It suggests that applying the ratio β_{ij} produces good solutions in some cases.

Table 19 presents the comparison between the optimal solution and the compound heuristic solutions to analyze the performances of the heuristic algorithms. This table shows the makespan for the optimal solution, makespan for the compound heuristic solution and the percentage deviation of makespan between the optimal solution and the compound heuristic solution. To calculate makespan, the loading time and unloading time of each type of product need to be known. Delay time for truck changes and the moving time of products from the receiving dock to the shipping dock should also be known. For all twenty sets of test problems, it is assumed that the loading time and unloading time are the same for all products and it takes one unit of time. Additionally, it is assumed that truck change time takes 75 units of time and travel time of products from the receiving dock to the shipping dock takes 100 units of time. With the above information, makespan is calculated as given in *equation (5-22)* and presented in Table 19.

Table 19. Makespans and Percentage Deviations of Makespan for the *Case 2* Problem

Problem Number	Makespan for Optimal Solution	Makespan for Compound Heuristic Solution	Percentage Deviation of Makespan
1	1840	1915	4.08%
2	1880	2030	7.98%
3	1515	1515	0.00%
4	2225	2375	6.74%
5	1810	1810	0.00%
6	1870	1870	0.00%
7	1830	1905	4.10%
8	1815	1815	0.00%
9	1825	1825	0.00%
10	1705	1780	4.40%
11	2470	2545	3.04%
12	3100	3250	4.84%
13	2910	2985	2.58%
14	2830	3055	7.95%
15	3030	3330	9.90%
16	2915	2915	0.00%
17	2030	2030	0.00%
18	2995	3070	2.50%
19	2945	3095	5.09%
20	3395	3620	6.63%

Percentage deviation of makespan between the optimal solution and the compound heuristic solutions is calculated as follows:

$$\begin{pmatrix} \textit{Percentage Deviation} \\ \textit{of Makespan (\%)} \end{pmatrix} = \frac{\begin{pmatrix} \textit{Makespan for Compound} \\ \textit{Heuristic Solution} \end{pmatrix} - \begin{pmatrix} \textit{Makespan for} \\ \textit{Optimal Solution} \end{pmatrix}}{\textit{Makespan for Optimal Solution}} \times 100 \quad (5\text{-}30)$$

As can be seen in Table 19, the range of percentage deviation for makespan in the twenty problem sets is 0%-9.90%. The overall average percentage deviation for makespan is 3.49%. The above analysis implies that the solutions found from the compound heuristic algorithm are very close to the optimal solutions.

In order to apply the complete enumeration method and the tabu search method to minimize the mean flow time for the *Case 2* problem, where the matching pairs and product routing are known from the previous solution, the minimum number of matching pairs which are obtained after applying the second mathematical model and presented in Appendix C were used. After applying the complete enumeration method and the tabu search method, the solutions are obtained and are as presented in Table 20. All algorithms were implemented on a personal computer (Intel Pentium Pro Microprocessor 200MHz) and the execution times were recorded for the problems. Because of the computational time, the complete enumeration method is only applied to eleven test problems whose number of matching pairs are less than or equal to twelve matching pairs. For one of the two problems with twelve matching pairs, it took more than seven hours to find the optimal solution using the complete enumeration method. The optimal receiving and shipping truck sequences for the eleven problems whose solutions were obtained from the complete enumeration method are presented in Appendix D. Appendix E presents the receiving truck and shipping truck sequences obtained from the tabu search method. As can be seen from Table 20, the tabu search found the optimal solution in all eleven test problems. The time required for finding the solutions by the tabu search ranged from 5 to 19 seconds and the computational time did not change significantly as the problem sizes increased in the tabu search. On the other hand, the solution time increased exponentially in the complete enumeration method as the problem sizes increased. The procedure behaved as expected.

Table 20. The Mean Flow Time for Complete Enumeration Solution and Tabu Solution for the *Case 2* Problem

Problem Number	Complete Enumeration Method				Tabu Search Method	
	Optimal Solution	Average Solution	Worst Solution	Elapsed Time	Tabu Solution	Elapsed Time
1	466.556	856.704	1263.000	17 min. 3sec.	466.556	5.823 sec.
2	482.000	864.830	1193.444	16 min. 54 sec.	482.000	5.823 sec.
3	623.833	857.583	1068.000	1.251 sec.	653.833	4.450 sec.
4	N/A*				668.500	10.179 sec.
5	565.625	949.083	1319.750	16 min 30 sec.	565.625	6.966 sec.
6	600.000	941.163	1290.625	16 min 32 sec.	600.000	6.121 sec.
7	423.222	808.822	1134.667	18 min. 2 sec.	423.222	6.084 sec.
8	590.000	987.958	1328.750	3 hrs. 37 min.	590.000	9.191 sec.
9	597.250	1020.608	1391.750	7 hrs. 48 min.	597.250	7.396 sec.
10	594.143	940.405	1263.857	1 min. 15 sec.	594.143	5.455 sec.
11	649.444	1199.889	1657.778	16 min. 52 sec.	649.444	6.606 sec.
12	N/A*				871.100	12.827 sec.
13	N/A*				740.364	17.222 sec.
14	N/A*				832.800	9.437 sec.
15	N/A*				679.455	8.153 sec.
16	N/A*				728.000	13.528 sec.
17	623.625	1075.375	1458.250	16 min. 43 sec.	623.625	6.485 sec.
18	N/A*				674.667	18.139 sec.
19	N/A*				828.700	11.848 sec.
20	N/A*				785.750	15.041 sec.

* N/A : Cannot find the solution because of computational time.

5.4 Conclusions

Three solution approaches have been developed to solve the cross docking problem for the *Case 2* model. A mixed integer programming model (Model I) to minimize the makespan of a cross docking operation was developed as the first approach. The second approach also employed another mathematical model. The second mathematical model (Model II) is an integer programming model whose objective is to minimize the number of matching pairs of the receiving and shipping trucks. By changing the objective, the number of variables and constraints in the problem is decreased drastically. Both mathematical models were able to find the global solution to the problem but are ineffective for solving medium to large size problems because of their intense computational requirements. Therefore, to improve solution efficiency, heuristic algorithms were developed.

The third approach used heuristic algorithms. Although the heuristics were able to find solutions to the problems rather quickly, no optimality is guaranteed. Six heuristic algorithms were developed and tested for the *Case 2* problem. Heuristic algorithms 1, 2 and 3 follow the same format except that they employ different criterion for selecting the best matching pair of the receiving and shipping trucks. In each iteration, the best matching pair is chosen based on the selection criterion. Heuristic algorithms 4, 5 and 6 are the modified versions of heuristic algorithms 1, 2, and 3, respectively. They use the same criteria as the first three algorithms, but they also have a condition for priority assignment. In a given iteration, if there are multiple pairs that satisfy the priority condition, then heuristic algorithms 1, 2 or 3 is applied only to the pairs that satisfy the priority condition. Thereafter, the best matching pair is chosen among those pairs that satisfy the priority condition. If there are no pairs that satisfy the priority condition in a given iteration, the process automatically reverts to heuristic algorithms 1, 2 or 3. Of the six heuristic algorithms, heuristic algorithm 3 (Maximum fitness) performed the best based on the test problems. It found the best solutions among the six heuristic solutions in fifteen out of the twenty test problems. Heuristic algorithms 2 and 5 yielded the best results in some cases. Overall, the heuristic algorithms produced solutions that were close to the global optimal solutions.

For the *Case 2* problem, once the minimum number of matching pairs is found then makespan is the same regardless of the order of selection of the matching pairs. However, the mean flow time or staying time of the receiving and shipping trucks at the warehouse depends on the spotting sequences of the receiving and shipping trucks. Therefore, two approaches were developed to find the best spotting sequences of the receiving and shipping trucks with the objective of minimizing the mean flow time for all trucks, where the number of matching pairs is as obtained from the complete enumeration method, Model II, heuristic, or the tabu search method, depending on which minimum matching pair procedure is used.

Using the complete enumeration method, the optimal solutions are found in eleven test problems among the twenty test problems because of computational time. It took more than seven hours to find the solution for one of the twelve matching pair problems. On the other hand, the tabu search found the solution very quickly. It found the solution within twenty seconds in all twenty test problems. The performance of the tabu search was very good. The tabu search found the optimal solution in all eleven problems whose optimal solutions were known and found by the complete enumeration method. To reduce the computational time of the tabu search, the net change of the flow time was developed and used instead of calculating the mean flow time for each adjacent neighborhood.

CHAPTER 6. *CASE 3* – CROSSDOCKING MODEL WITH TEMPORARY STORAGE AND DOCK REPEAT TRUCK HOLDING PATTERN

6.1 Model Descriptions

In the third case of the cross docking problem studied in this research, it is assumed that there is temporary storage in front of the shipping dock and that both the receiving trucks and the shipping trucks can intermittently move in and out of the dock during the time intervals between their task execution. After a receiving truck unloads some of its products for a certain shipping truck, one of two choices can be made; either more products are unloaded from the current receiving truck and sent to the temporary storage, or the current receiving truck is moved out from the receiving dock and another receiving truck is sent to the receiving dock to unload its products. This operation plan can be similarly applied to the shipping truck. Suppose a shipping truck loads some of its needed products from a certain receiving truck or temporary storage and no products at the shipping dock are needed for the current shipping truck. Then, the current shipping truck either waits until its needed products arrive at the shipping dock or is allowed to move out from the shipping dock and another shipping truck is sent to the shipping dock to load its needed products.

The objective of the *Case 3* problem is the same as in *Case 1* or *Case 2* problems. It is to find the best sequence for truck spotting for both the receiving and shipping trucks to minimize total operation time or to maximize the throughput of the cross docking system. Additionally, the solution needs to show the product routings or the product assignments. In other words, the solution needs to show how many products move from a certain receiving truck to a certain shipping truck as well as what types of products move between them. The solution also needs to show whether the products move directly from a receiving truck to a shipping truck or pass through temporary storage during a transition.

In the *Case 3* problem, there are two types of delay times. The first type of delay time occurs when there is a shipping truck change. The second type of delay time occurs when the current shipping truck does not load any products from a certain receiving truck or temporary storage and waits until its needed products arrive at the shipping dock. The change of receiving trucks or the unloading of products from a receiving truck and sending the products to temporary storage may also cause the second type of delay time. For *Case 3* problem, makespan is equal to the total delay time plus the total unloading or loading time of all products. Since the latter is a constant in any schedule, the minimization of the makespan is equivalent to the minimization of the total delay time in *Case 3* problem.

The *Case 3* problem has the following characteristics:

1. If delay time for a truck change, *D*, is relatively larger than the average length of time required to unload one batch of products from a receiving truck, it is preferable to decrease the number of truck changes rather than decrease the number of products that pass through the temporary storage. A batch of products is defined as a set of products unloaded consecutively from a given receiving truck without any time delay or interruption. It is because the shipping truck may need to wait at the dock while the receiving truck is unloading the products and sending the unloaded products to temporary storage. If *D* is very large, it is better to hold a receiving truck at the dock to unload its products and sending the unloaded products to the temporary storage instead of frequently changing the receiving and shipping trucks at the docks. Therefore, the approach adopted in solving the problem is to first minimize the number of truck changes, and then minimize the number of products that pass through temporary storage. This situation is the same as that of *Case 1* problem. The objective of *Case 1* problem is to minimize the number of products that pass through the temporary storage. Because in the *Case 1* problem a receiving truck or shipping truck can only visit the dock once, the optimum strategy is to fix the number of truck changes to the minimum.
2. On the other hand, if the average time required to unload one batch is relatively larger than *D*, it is preferable to decrease the number of products that pass through the temporary storage rather than decrease the number of truck changes. If the average time required to unload one batch is very large, it is preferable to incur more visits and consequently more changes of the receiving and shipping trucks at the docks than to send products to temporary storage. In this situation, the *Case 3* problem will turn out to behave like that of the *Case 2* problem. The objective of *Case 2* problem is to minimize the number of truck changes. No products are allowed to pass through temporary storage in the *Case 2* problem.

From the above characteristics, it is obvious that the solution of *Case 3* problem depends on the values of D and u_k, where u_k is defined as the unloading time for one unit of product type k from a receiving truck, because the time required to unload a batch is directly related to u_k. In other words, different solutions can be obtained for the same truck and product characteristics depending on the values of D and u_k.

6.2 Model Developments

To solve the cross docking problem for *Case 3* problem, two approaches were developed. A mathematical model was developed as the first approach. Even though the mathematical model for *Case 3* problem can be developed, it is difficult to solve because the number of variables and constraints grow exponentially as the number of receiving trucks, the number of shipping trucks, and the number of product types increase. As a result, a second approach was developed to solve the problems. The second approach employs heuristic algorithms. The heuristics were able to obtain solutions to the problem very quickly, except that no optimality is guaranteed.

6.2.1 Mathematical Model

The following mathematical model was developed to find the optimal solution that minimizes the makespan for *Case 3* problem.

6.2.1.1 Notations

The following notations are used for the mathematical model:

Continuous Variables:

T = Makespan,

U_{ij} = Time at which the variable t_{ij} transferring from receiving truck i to shipping truck j starts to unload from receiving truck i onto the receiving dock,

L_{ij} = Time at which the variable t_{ij} transferring from receiving truck i to shipping truck j finished loading from the shipping dock into shipping truck j,

Integer Variables:

x_{ijk} = Number of units of product type k which transfer from receiving truck i to shipping truck j,

t_{ij} = Total number of units of products which transfer from receiving truck i to shipping truck j, where $\left(t_{ij} = \sum_{k=1}^{N} x_{ijk}\right)$,

Binary Variables:

$$v_{ij} = \begin{cases} 1, & \text{If any products transfer from receiving truck } i \text{ to shipping truck } j \\ 0, & \text{Otherwise} \end{cases},$$

$$p_{iji'j'} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ immediately or directly precedes the vairable } t_{i'j'} \text{ in the receiving sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$p_{00i'j'} = \begin{cases} 1, & \text{If the variable } t_{i'j'} \text{ is placed at the first position in the receiving sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$p_{ij00} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ is placed at the last position in the receiving sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$q_{iji'j'} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ immediately or directly precedes the variable } t_{i'j'} \text{ in the shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$q_{00i'j'} = \begin{cases} 1, & \text{If the variable } t_{i'j'} \text{ is placed at the first position in the shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$q_{ij00} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ is placed at the last position in the shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

Data:

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k which is initially loaded in receiving truck i,

s_{jk} = Number of units of product type k which is initially needed for shipping truck j,

D = Delay time for truck change,

V = Moving or travel time of products from the receiving dock to the shipping dock,

M = Big number.

6.2.1.2 Mixed Integer Programming Model

For the mathematical model of the *Case 3* problem, it is assumed that the unloading time from a receiving truck and the loading time into a shipping truck are the same for all products and it takes one unit of time for one unit of products. Additionally, it is assumed it takes one unit of time to unload one unit of any product from a conveyor to the temporary storage or loaded from the temporary storage into a shipping truck. With the above assumptions, the following mixed integer programming model was developed for the *Case 3* problem.

<u>**Mathematical Model for the *Case 3* Problem**</u>

Min $\qquad T$

Subject to

$$T \geq L_{ij}, \quad for\ all\ i, j \tag{6–1}$$

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \quad for\ all\ i, k \tag{6–2}$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \quad for\ all\ j, k \tag{6–3}$$

$$\sum_{k=1}^{N} x_{ijk} = t_{ij}, \quad for\ all\ i, j \tag{6–4}$$

$$t_{ij} \leq M\, v_{ij}, \quad for\ all\ i, j \tag{6–5}$$

$$v_{ij} = \sum_{i'=1}^{R} \sum_{j'=1}^{S} p_{iji'j'} + p_{ij00}, \quad for\ all\ i, j \tag{6–6}$$

$$v_{i'j'} = \sum_{i=1}^{R} \sum_{j=1}^{S} p_{iji'j'} + p_{00i'j'}, \quad for\ all\ i', j' \tag{6–7}$$

$$v_{ij} = \sum_{i'=1}^{R} \sum_{j'=1}^{S} q_{iji'j'} + q_{ij00}, \quad for\ all\ i, j \tag{6–8}$$

$$v_{i'j'} = \sum_{i=1}^{R}\sum_{j=1}^{S} q_{iji'j'} + q_{00i'j'}, \quad for\ all\ i', j' \tag{6–9}$$

$$\sum_{i'=1}^{R}\sum_{j'=1}^{S} p_{00i'j'} = 1, \tag{6–10}$$

$$\sum_{i=1}^{R}\sum_{j=1}^{S} p_{ij00} = 1, \tag{6–11}$$

$$\sum_{i'=1}^{R}\sum_{j'=1}^{S} q_{00i'j'} = 1, \tag{6–12}$$

$$\sum_{i=1}^{R}\sum_{j=1}^{S} q_{ij00} = 1, \tag{6–13}$$

$$p_{ijij} = 0, \quad for\ all\ i, j \tag{6–14}$$

$$q_{ijij} = 0, \quad for\ all\ i, j \tag{6–15}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} - M(1 - p_{iji'j'}), \quad for\ all\ i, j, i', j'\ and\ where\ i = i' \tag{6–16–a}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} + D - M(1 - p_{iji'j'}), \quad for\ all\ i, j, i', j'\ and\ where\ i \neq i' \tag{6–16–b}$$

$$L_{ij} \geq U_{ij} + V + t_{ij}, \quad for\ all\ i, j \tag{6–17}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} - M(1 - q_{iji'j'}), \quad for\ all\ i, j, i', j'\ and\ where\ j = j' \tag{6–18–a}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} + D - M(1 - q_{iji'j'}), \quad for\ all\ i, j, i', j'\ and\ where\ j \neq j' \tag{6–18–b}$$

$$all\ variables \geq 0.$$

Constraint (6–1) ensures that makespan is greater than or equal to the time the last product is loaded onto the last scheduled shipping truck. *Constraint (6–2)* ensures that the total number of units of product type k that transfer from receiving truck i to all shipping trucks is exactly the same as the number of units of product type k which is initially loaded in receiving truck i. Similarly, *constraint (6–3)* ensures that the total number of units of product type k that transfer from all receiving trucks to shipping truck j is exactly the same as the number of units of product type k which is initially needed for shipping truck j. *Constraint (6–4)* defines the t_{ij} variables which is used in *constraints (6–16)* to *(6–18)* in order to calculate the unloading and loading times. *Constraint (6–5)* just enforces the correct relationship between the t_{ij} variables and the v_{ij} variables.

Constraint (6–6) ensures that only one of the t_{ij} variables can immediately or directly precede another $t_{i'j'}$ variable in the receiving sequence when $v_{ij} = 1$. *Constraint (6–7)* ensures that only one of the $t_{i'j'}$ variables can immediately or directly follow another t_{ij} variable in the receiving sequence when $v_{i'j'} = 1$. Similar to *constraints (6–6)* and *(6–7)*, *constraint (6–8)* ensures that only one of the t_{ij} variables can immediately precede another $t_{i'j'}$ variable in the shipping sequence when $v_{ij} = 1$ and *constraint (6–9)* ensures that only one of the $t_{i'j'}$ variables can immediately follow another t_{ij} variable in the shipping sequence when $v_{i'j'} = 1$.

Constraint (6–10) ensures only one of the $t_{i'j'}$ variables can be placed at the first position of the receiving sequence. *Constraint (6–11)* ensures only one of the t_{ij} variables can be placed at the last position of the receiving sequence. Similarly, *constraints (6–12)* and *(6–13)* ensures only one of the $t_{i'j'}$ variables can be placed at the first position and only one of the t_{ij} variables can be placed at the last position of the shipping sequence, respectively. *Constraints (6-14)* and *(6-15)* ensure that there are no consecutive sequences that transfer products from the same receiving truck to the same shipping truck.

Constraints (6–16–a) and *(6–16–b)* make a valid sequence of unloading times for the t_{ij} variables, based on their order. If there is no receiving truck change between the consecutive unloading sequences (in case of $i = i'$), *constraint (6–16–a)* is applied. However, if there is a change of receiving trucks between the consecutive unloading sequences (in case of $i \neq i'$), the delay time for receiving truck change must be considered, thus *constraint (6–16–b)* is applied.

Constraint (6–17) establishes the proper relationship between the variables U_{ij} and L_{ij}. Finally, *constraints (6–18–a)* and *(6–18–b)* ensure a valid sequence for the loading times of the t_{ij} variables, based on their order. If there is no shipping truck change between the consecutive loading sequences (in case of $j = j'$), *constraint (6–18–a)* is applied However, if there is a change of shipping trucks between the consecutive loading sequences (in case of $j \neq j'$), then the delay time for shipping truck change must be considered, thus *constraint (6–18–b)* is applied.

The number of decision variables for this integer programming model is *RS(2RS+N+8)+1*. The decision variables consist of *RS(2RS+5)* of binary variables, *RS(N+1)* of integer variables and *(2RS+1)* of continuous variables. The number of constraints is *2RS(RS+5)+N(R+S)+4*, including *RS(2RS+3)* of inequality constraints and *(7RS+RN+SN+4)* of equality constraints. The number of decision variables and constraints for some representative values of *R*, *S*, and *N* is illustrated in Table 21.

As can be seen in Table 21, the number of variables and constraints in the mathematical model grow exponentially based on the number of receiving trucks, the number of shipping trucks, and the number of product types involved. Because the computational intensity of the mathematical model is too high and therefore makes the approach impractical to use, the heuristic algorithm was developed to solve the *Case 3* problem.

6.2.1.3 Interpretation of the Solution

First, the receiving sequence of the t_{ij} variables can be found from the $p_{00i'j'}, p_{iji'j'}$, and p_{ij00} variables. From the receiving sequence of the t_{ij} variables, the receiving truck spotting sequence can be identified. Similarly, the shipping sequence of the t_{ij} variables can be found from the $q_{00i'j'}, q_{iji'j'}$, and q_{ij00} variables. From the shipping sequence of the t_{ij} variables, the shipping truck spotting sequence can be identified. The number of products unloaded and loaded can be found from the x_{ijk} variables. The variable T represents the makespan for the total cross docking operation. Detailed information about the unloading time and the loading time of the t_{ij} variables can be found from the U_{ij} and L_{ij} variables.

Table 21. The Number of Decision Variables and Constraints for Some Representative Values of *R*, *S*, and *N* of the Mathematical Model for the *Case 3* Problem

R	*S*	*N*	Decision Variables				Constraints		
			Binary	**Integer**	**Continuous**	**Total**	**Equality**	**Inequality**	**Total**
3	3	8	207	81	19	307	115	189	304
4	4	5	592	96	33	721	156	560	716
5	5	10	1375	275	51	1701	279	1325	1604
5	6	8	1950	270	61	2281	302	1890	2192
6	6	9	2772	360	73	3205	364	2700	3064

6.2.2 Heuristic Method

As mentioned at the beginning of this chapter, there are two types of idle times for the *Case 3* problem; 1) truck change time, and 2) the idle time of a shipping truck at the shipping dock while it waits for the arrival of products from the receiving dock. For the *Case 3* problem, one of two types of decisions can be made after products, which are transferred from a current receiving truck to a certain shipping truck, are unloaded from the current receiving truck. The first type of decisions is to change the current receiving truck to another receiving truck. The second type of decisions is to unload more products from the current receiving truck and send the unloaded products into temporary storage.

Similarly, one of two types of decisions can be made after a current shipping truck loads all of its needed products available at the shipping dock; 1) change the current shipping truck to another shipping truck, and 2) the current shipping truck is allowed to wait at the shipping dock until its other needed products arrive from the receiving dock. Depending on the decision made at each decision point, delay time may be added to increase the makespan. The heuristic algorithm for the *Case 3* problem must be able to choose the schedule that adds the smallest idle time to makespan at each decision point.

The heuristic algorithm developed for the *Case 3* problem consists of two phases. In *Phase I*, the product routing is decided. The initial receiving and shipping truck sequences are also created in *Phase I*. In the schedule of *Phase I*, no products are sent to temporary storage. In *Phase II*, a check is made to determine whether the makespan is decreased by changing the current receiving truck at the dock for another receiving truck or keeping the current receiving truck at the dock to continue unloading its items and sending the items to temporary storage. If a certain condition that decreases makespan by sending products to temporary storage is met, the schedule is modified to unload more products from the current receiving truck and moving the unloaded items to temporary storage instead of changing the receiving truck. *Phase II* is continued until the schedules do not satisfy any conditions that decrease makespan. Throughout this section, it is assumed that all unloading and loading times are the same for all products and that this time is one unit long in duration for one unit of any products.

6.2.2.1 Notations

The following notations are used in this section:

R : Number of receiving trucks in the set,

S : Number of shipping trucks in the set,

N : Number of product types in the set,

r_{ik} : Number of units of product type k which is initially loaded in receiving truck i,

s_{jk} : Number of units of product type k which is initially needed for shipping in truck j,

D : Delay time for truck change,

V : Moving time of products from the receiving dock to the shipping dock,

t^r_i : Receiving truck i,

t^s_j : Shipping truck j,

A^s_j : Set of associate receiving trucks for shipping truck j,

T^r : The ordered set of scheduled receiving trucks,

T^s : The ordered set of scheduled shipping trucks,

T^p : The ordered set of product routing based on the receiving and shipping truck sequences,

α_{ij} : Product routing presented in set T^p (α_{ij} represents the products that are transferred from receiving truck i to shipping truck j),

T^n : The ordered set of the number of products transferred from a receiving truck to a shipping truck corresponding to the product routing α_{ij} in T^p,

β_{ij} : Total number of units of products transferred from receiving truck i to shipping truck j corresponding to the product routing α_{ij},

F^r : The ordered set of completion time of receiving truck i corresponding to product routing α_{ij} in set T^p,

γ_{ij} : Completion time of the receiving truck i corresponding to product routing α_{ij} in set T^p,

F^s : The ordered set of leaving time of the shipping truck corresponding to the shipping truck sequence in set T^s,

δ_i : Leaving time of the shipping truck i corresponding to the shipping truck sequence in set T^s.

6.2.2.2 *Phase I* of Heuristic Algorithm for the *Case 3* Problem

At each iteration of *Phase I*, the first step is to find the best associate receiving trucks for each unscheduled shipping truck. Then the shipping truck that has the smallest number of associate receiving trucks is selected as the next scheduled shipping truck because it will minimize delay time for receiving truck changes. The selected shipping truck and its associate receiving trucks are scheduled in the shipping truck sequence and the receiving truck sequence, respectively.

Once a shipping truck and its associate receiving trucks are scheduled, the remaining number of products in the receiving trucks is updated. Next, for each unscheduled shipping truck, a new set of its associate receiving truck is formed from the updated receiving truck list. Again, the shipping truck that has the smallest number of associate receiving trucks is selected and scheduled. Once a shipping truck and its associate receiving trucks are selected and scheduled, the remaining number of products in the receiving trucks is again updated. The process of selection, scheduling and updating is continued until all trucks are scheduled. When all receiving and shipping trucks are scheduled, *Phase I* is terminated and *Phase II* is started. The heuristic algorithm for *Phase I* is presented below.

<u>*PHASE I* OF THE HEURISTIC ALGORITHM FOR THE *CASE 3* PROBLEM</u>

<u>STEP 1</u>

Initialize sets T^r, T^s, T^p and T^n. $T^r = \varnothing$, $T^s = \varnothing$, $T^p = \varnothing$ and $T^n = \varnothing$.

<u>STEP 2</u>

For each unscheduled shipping truck $t^s_j \in T^s$, find its best associate receiving trucks A^s_j and product routing α_{ij}, where $t^r_i \in A^s_j$, using one of the following strategies:

Strategy 1 – Maximum flow between the receiving truck and the shipping truck. (*Strategy 1* is similar to *Heuristic Algorithm 1* for the *Case 2* problem.)

Strategy 2 – Maximum ratio between the receiving truck and the shipping truck. (*Strategy 2* is similar to *Heuristic Algorithm 2* for the *Case 2* problem.)

Strategy 3 – Maximum fitness between the receiving truck and the shipping truck. (*Strategy 3* is similar to *Heuristic Algorithm 3* for the *Case 2* problem.)

The procedure of forming the best associate receiving trucks A^s_j of an unscheduled shipping truck t^s_j involves the sequential selection of one of the receiving trucks based on one of the above selection criteria in each iteration. After the best receiving truck is selected in each iteration, the remaining number of products in the shipping truck is updated. The above procedure is continued until the shipping truck loads all of its needed products from its best associate receiving trucks. The selection procedure for finding the best associate receiving trucks is similar to the selection procedure for the *Case 1* problem presented in Section 4.2.3.2.

<u>*STEP 3*</u>

Choose the shipping truck that has the smallest number of associate receiving trucks. If there is a tie, choose the shipping truck that needs the largest number of products.

<u>*3a*</u> Place the selected shipping truck, $t^s_{j^*}$ at the end of the sequence in set T^s.

<u>*3b*</u> Schedule the best associate receiving trucks of the selected shipping truck, $A^S_{j^*} = \{t^r_{[1]}, t^r_{[2]}, \ldots, t^r_{[k]}\}$ at the end of sequence in set T^r.

<u>*3c*</u> Place the product routing, $\{\alpha_{[1]j^*}, \alpha_{[2]j^*}, \ldots, \alpha_{[k]j^*}\}$ at the end of sequence in set T^p.

<u>*3d*</u> Place the total number of products transferred corresponding to $\alpha_{[i]j^*}$, where $1 \le i \le k$, in set T^p (i.e. $\{\beta_{[1]j^*}, \beta_{[2]j^*}, \ldots, \beta_{[k]j^*}\}$) at the end of sequence in set T^n.

<u>*STEP 4*</u>

Update the remaining number of products in the receiving trucks. If there is any unscheduled shipping truck, go to *Step 2*. Otherwise, stop *Phase I*. The solution for *Phase I* is found. The solution shows the following four sequences:

1. Receiving truck sequence T^r
2. Shipping trucks sequence T^s
3. Information for product routing T^p
4. Total number of products transferred from a receiving truck to a shipping truck T^n.

Go to *Phase II*.

Figure 15 describes the algorithmic steps of the heuristic algorithm of *Phase I*. To illustrate *Phase I* of the *Case 3* problem, consider Example 4 as described below. Example 4 has five receiving trucks, four shipping trucks and six product types as presented in Table 22. Example 4 is the same example as test problem set 7 in Appendix B. It is assumed that truck change time takes 20 units of time and item moving time from the receiving dock to the shipping dock takes 10 units of time. Suppose *Strategy 1*, "Maximum flow between the receiving and shipping trucks", is adapted for this example. Then, *Phase I* proceeds as follows:

After sets T^r, T^s, T^p and T^n are initialized in *Step 1* of *Phase I*, *Step 2* is to find the best associate receiving trucks for each unscheduled shipping truck. The procedure to find the best associate receiving trucks for shipping truck 1 is presented in Table 23. In the first iteration, β_{11} = 42 because 14 units of product type 1 and 28 units of product type 3 can be moved from receiving truck 1 to shipping truck 1. Following the procedure, the values of β_{21}, β_{31}, β_{41} and β_{51} are calculated. After all values of β_{11}, β_{21}, β_{31}, β_{41} and β_{51} are found, receiving truck 5 is selected in the first iteration because it has the highest value which is 131. From receiving truck 5 to shipping truck 1, 36 units of product type 3 and 95 units of product type 6 can be transferred. After receiving truck 5 is selected in the first iteration, the remaining products in the shipping truck are updated and the procedure is repeated.

In Iteration 2, receiving truck 2 is selected because it has the highest value of β_{21} = 50. After receiving truck 2 is selected in the second iteration, the loading requirement for shipping truck 1 is fully satisfied. Therefore, the best associate receiving trucks for shipping truck 1 are found and they are receiving trucks 5 and 2. A similar procedure is applied for shipping trucks 2, 3 and 4 in order to find their best associate receiving trucks. After applying the procedure to all remaining shipping trucks, the following result is obtained.

$A^S_1 = \{t^r_5, t^r_2\}$.

$A^S_2 = \{t^r_3, t^r_5, t^r_1\}$.

$A^S_3 = \{t^r_1, t^r_4, t^r_2, t^r_5\}$.

$A^S_4 = \{t^r_4\}$.

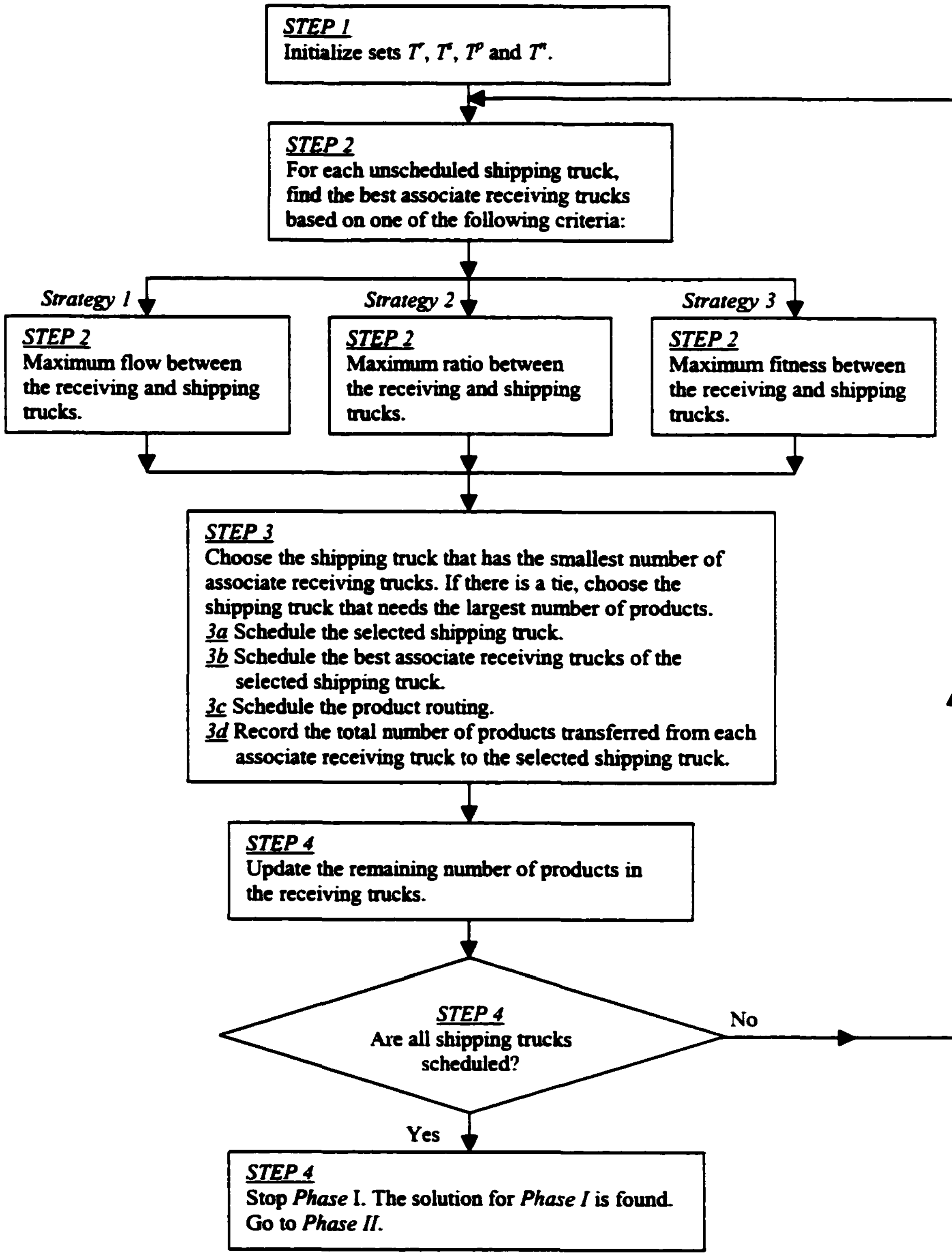


Figure 15. *Phase I* of the Heuristic Algorithm for the *Case 3* Problem

Table 22. Example Set 4 to Illustrate the Heuristic Algorithm for the *Case 3* Problem

Receiving Truck

Truck	Product	Quantity
1	1	14
	2	69
	3	28
	5	69
2	1	50
	4	40
	6	70
3	2	190
4	1	23
	3	115
	5	92
5	3	44
	5	66
	6	110

Shipping Truck

Truck	Product	Quantity
1	1	50
	3	36
	6	95
2	2	194
	3	53
	5	62
	6	74
3	1	37
	2	65
	3	18
	4	40
	5	165
	6	11
4	3	80

Table 23. Procedure for Finding the Best Associate Receiving Trucks for Shipping Truck 1

Iteration	Receiving Truck 1	Receiving Truck 2	Receiving Truck 3	Receiving Truck 4	Receiving Truck 5	Selected Receiving Truck
1	$\beta_{11} = 42$	$\beta_{21} = 120$	$\beta_{31} = 0$	$\beta_{41} = 59$	$\beta_{51} = 131$	t^r_5
2	$\beta_{11} = 14$	$\beta_{21} = 50$	$\beta_{31} = 0$	$\beta_{41} = 23$	$\beta_{51} = 0$	t^r_2

Because all unscheduled shipping trucks have found their best associate receiving trucks, then *Step 3* of *Phase I* can be started at this point. In *Step 3*, shipping truck t^s_4 and its associate receiving truck t^r_4 are selected because shipping truck t^s_4 has the smallest number of associate receiving trucks. Now, the remaining number of products in receiving and shipping trucks is updated as shown in Table 24 in *Step 4*.

Table 24. The Remaining Products after Shipping Truck 4 and Its Associate Receiving Truck 4 are selected in the First Iteration

Receiving Truck

Truck	Product	Quantity
1	1	14
	2	69
	3	28
	5	69
2	1	50
	4	40
	6	70
3	2	190
4	1	23
	3	35 (Updated)
	5	92
5	3	44
	5	66
	6	110

Shipping Truck

Truck	Product	Quantity
1	1	50
	3	36
	6	95
2	2	194
	3	53
	5	62
	6	74
3	1	37
	2	65
	3	18
	4	40
	5	165
	6	11

* Shipping truck 4 is removed because it is already scheduled and all its needs have been met.

Because shipping trucks t^s_1, t^s_2 and t^s_3 are not yet scheduled, the procedure is continued by going back to *Step 2*. In *Step 2*, for unscheduled shipping trucks t^s_1, t^s_2 and t^s_3, a new set of its associate receiving truck A^S_1, A^S_2 and A^S_3 are formed from the updated receiving truck list in Table 24. Again, the shipping truck that has the smallest number of associate receiving trucks is selected and scheduled. The process of selection, scheduling and updating is continued until all trucks are scheduled. Tables 25 and 26 show the solution for the *Phase I* algorithm sequenced according to the order of selection of the shipping trucks and their associate receiving trucks. At the end of *Phase I*, the complete solution for *Phase I* is presented as follows:

$$\begin{array}{lllll} & \text{Associate Receiving} & \text{Associate Receiving} & \text{Associate Receiving} & \text{Associate Receiving} \\ & \text{Trucks for } t^s_4 & \text{Trucks for } t^s_1 & \text{Trucks for } t^s_3 & \text{Trucks for } t^s_2 \\ T^r = \{ & t^r_4, & t^r_5, t^r_2, & t^r_1,\ t^r_4,\ t^r_2,\ t^r_5, & t^r_3,\ t^r_5,\ t^r_2,\ t^r_4, t^r_1\}. \\ T^s = \{ & t^s_4, & t^s_1, & t^s_3, & t^s_2 \quad \}. \\ T^p = \{ & \alpha_{44}, & \alpha_{51}, \alpha_{21}, & \alpha_{13}, \alpha_{43}, \alpha_{23}, \alpha_{53}, & \alpha_{32}, \alpha_{52}, \alpha_{22}, \alpha_{42}, \alpha_{12}\}. \\ T^n = \{ & 80, & 131, 50, & 166, 115, 51,\ 4, & 190, 85,\ 59,\ 35,\ 14\}. \end{array} \qquad (6\text{-}19)$$

Table 25. The Selected Sequences of the Shipping Trucks and Their Associate Receiving Trucks after Applying *Phase I* for the *Case 3* Problem

Iteration	Shipping Truck	Associate Receiving Trucks
1	t^s_4	$A^S_4 = \{t^r_4\}$
2	t^s_1	$A^S_1 = \{t^r_5, t^r_2\}$
3	t^s_3	$A^S_3 = \{t^r_1, t^r_4, t^r_2, t^r_5\}$
4	t^s_2	$A^S_2 = \{t^r_3, t^r_5, t^r_2, t^r_4, t^r_1\}$.

Table 26. Product Routing between Receiving and Shipping Trucks after Applying the *Phase I* Algorithm for the *Case 3* Problem

Receiving Truck	Shipping Truck	Total Number of Products Transferred
t^r_4	t^s_4	80
t^r_5	t^s_1	131
t^r_2	t^s_1	50
t^r_1	t^s_3	166
t^r_4	t^s_3	115
t^r_2	t^s_3	51

Receiving Truck	Shipping Truck	Total Number of Products Transferred
t^r_5	t^s_3	4
t^r_3	t^s_2	190
t^r_5	t^s_2	85
t^r_2	t^s_2	59
t^r_4	t^s_2	35
t^r_1	t^s_2	14

6.2.2.3 *Phase II* of Heuristic Algorithm for the *Case 3* Problem

At the beginning of *Phase II*, the initial sequences for the receiving and shipping trucks are known from the solution of *Phase I*. Also known from *Phase I* are the product routing and the number of products transferred from a scheduled receiving truck to a scheduled shipping truck. The solution of *Phase I* does not send any products to temporary storage. Instead of sending products to temporary storage, receiving truck changes are employed. In *Phase II*, a search is carried out to decrease the makespan by sending products to temporary storage instead of changing the receiving trucks.

The heuristic algorithm for *Phase II* does not change the sequence of the shipping trucks. Therefore, the shipping truck sequence of *Phase II* remains the same as the shipping sequence of *Phase I*. On the other hand, the rest of the sequences such as the receiving truck sequence (i.e. set T^r), the sequence for product routing (i.e. set T^p) and the sequence for the number of products transferred (i.e. set T^n) are modified if the modified sequences decrease makespan. This implies there is the possibility of decreasing makespan by changing the receiving truck sequence and the product routing in *Phase II*.

In *Phase II*, two conditions that can decrease makespan are identified. For the first condition, consider two consecutively scheduled shipping trucks t^s_i and t^s_j in the shipping truck sequence and their associate receiving trucks in the receiving truck sequence are presented as follows:

$$
\begin{aligned}
T^r &= \{\ \dots,\quad \overbrace{, \dots, t^r_p, \dots,}^{\text{Associate Receiving Trucks for } t^s_i} \quad \overbrace{\dots, t^r_p, \dots,}^{\text{Associate Receiving Trucks for } t^s_j} \quad \dots\ \}. \\
T^s &= \{\ \dots,\quad t^s_i, \quad t^s_j, \quad \dots\ \}.
\end{aligned}
$$

Suppose the two consecutive shipping trucks, t^s_i and t^s_j, need the same associate receiving truck, t^r_p. It means that the receiving truck t^r_p needs to be scheduled twice at the dock for the two consecutively scheduled shipping trucks. Suppose receiving truck t^r_p unloads the products needed for shipping truck t^s_j immediately after it unloads the products needed for shipping truck t^s_i. If this occurs, then the truck changing time for receiving truck

t^r_p is avoided (i.e., saved) because receiving truck t^r_p only needs to come to the receiving dock once for the two consecutive shipping trucks, t^s_i and t^s_j. On the other hand, the departure time of shipping truck t^s_i may be delayed because shipping truck t^s_i may need to wait without being loaded with products while receiving truck t^r_p unloads products for shipping truck t^s_j. If this happens, the departure time of shipping truck t^s_i can be delayed up to the completion time for unloading the products which transfer from receiving truck t^r_p to shipping truck t^s_j, thus makespan can be increased by an amount of time equal to the time required to unload the products transferred to temprorary storage. From the above observation, it can be seen that if two consecutive shipping trucks, t^s_i and t^s_j, need the same associate receiving truck t^r_p and the total time required to unload products which transfer from receiving truck t^r_p to shipping truck t^s_j is less than the delay time for truck change, then the makespan can be decreased by unloading the products needed for shipping truck t^s_j immediately after unloading the products needed for shipping truck t^s_i. Based on the above characteristic, the first condition for modifying the schedules in *Phase II* is explained below.

Condition 1

Suppose that shipping trucks t^s_h, t^s_i and t^s_j are sequentially scheduled in the shipping truck sequence. If shipping trucks t^s_i and t^s_j have the same associate receiving trucks, then the sequences in sets T^r, T^p and T^n are modified based on one of the following situations:

1. Shipping trucks t^s_i and t^s_j have in common only one associate receiving truck t^r_p.
   i) If all of the following three conditions, *C1*, *C2* and *C3*, are satisfied, the sequences will appear as in sequence *(6-20)*.

      C1. $\beta_{pj} < D$.

      C2. Receiving truck t^r_p is the first scheduled receiving truck among the associate receiving trucks of shipping truck t^s_i.

      C3. Receiving truck t^r_p is the last scheduled receiving truck among the associate receiving trucks of shipping truck t^s_h.

$$
\begin{array}{llllll}
 & \overbrace{\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_h} & \overbrace{\qquad\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_i} & \overbrace{\qquad\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_j} & & \\
T^r = \{\ldots, & \ldots, t^r_p, & t^r_p, t^r_a, \ldots, t^r_b, & \ldots, t^r_c, t^r_p, t^r_d, \ldots, & \ldots\}. & \\
T^s = \{\ldots, & t^s_h, & t^s_i, & t^s_j, & \ldots\}. & \\
T^p = \{\ldots, & \ldots, \alpha_{ph}, & \alpha_{pi}, \alpha_{ai}, \ldots, \alpha_{bi}, & \ldots, \alpha_{cj}, \alpha_{pj}, \alpha_{dj}, \ldots, & \ldots\}. & \\
T^n = \{\ldots, & \ldots, \beta_{ph}, & \beta_{pi}, \beta_{ai}, \ldots, \beta_{bi}, & \ldots, \beta_{cj}, \underset{\substack{\uparrow \\ \beta_{pj} < D}}{\beta_{pj}}, \beta_{dj}, \ldots, & \ldots\}. & \text{(6-20)}
\end{array}
$$

Then sequences in sets T^r, T^p and T^n can be modified as follows to decrease makespan:

1. In set T^r, remove the associate receiving truck t^r_p of shipping truck t^s_j.
2. In set T^p, move the product routing α_{pj} to the next position of product routing α_{pi}.
3. In set T^n, move β_{pj} to the corresponding position of α_{pj} above.

The modified sequences in sets T^r, T^p and T^n will appear as in sequence *(6-21)*.

$$
\begin{array}{llllll}
 & \overbrace{\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_h} & \overbrace{\qquad\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_i} & \overbrace{\qquad\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_j} & & \\
T^r = \{\ldots, & \ldots, t^r_p, & t^r_p, \quad t^r_a, \ldots, t^r_b, & \ldots, t^r_c, \; t^r_d, \ldots, & \ldots\}. & \\
T^s = \{\ldots, & t^s_h, & t^s_i, & t^s_j, & \ldots\}. & \\
T^p = \{\ldots, & \ldots, \alpha_{ph}, & \alpha_{pi}, \alpha_{pj}, \alpha_{ai}, \ldots, \alpha_{bi}, & \ldots, \alpha_{cj}, \alpha_{dj}, \ldots, & \ldots\}. & \\
T^p = \{\ldots, & \ldots, \beta_{ph}, & \beta_{pi}, \beta_{pj}, \beta_{ai}, \ldots, \beta_{bi}, & \ldots, \beta_{cj}, \beta_{dj}, \ldots, & \ldots\}. & \text{(6-21)}
\end{array}
$$

ii) Otherwise (i.e. if any of the above three conditions *C1*, *C2* or *C3*, is not satisfied), the sequences may appear as in sequence *(6-22)*.

$$
\begin{array}{lllll}
 & \overbrace{\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_h} & \overbrace{\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_i} & \overbrace{\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_j} & \\
T^r = \{\ldots, & \ldots, & \ldots, t^r_a, t^r_p, t^r_b, \ldots, & \ldots, t^r_c, t^r_p, t^r_d, \ldots, & \ldots\}. \\
T^s = \{\ldots, & t^s_h, & t^s_i, & t^s_j, & \ldots\}. \\
T^p = \{\ldots, & \ldots, & \ldots, \alpha_{ai}, \alpha_{pi}, \alpha_{bi}, \ldots, & \ldots, \alpha_{cj}, \alpha_{pj}, \alpha_{dj}, \ldots, & \ldots\}. \\
T^n = \{\ldots, & \ldots, & \ldots, \beta_{ai}, \beta_{pi}, \beta_{bi}, \ldots, & \ldots, \beta_{cj}, \beta_{pj}, \beta_{dj}, \ldots, & \ldots\}.
\end{array}
\qquad (6\text{-}22)
$$

Then sequences in sets T^r, T^p and T^n can be modified as follows to decrease makespan:

a. In set T^r, remove the associate receiving truck t^r_p of shipping truck t^s_j. Move the associate receiving truck t^r_p of shipping truck t^s_i to the last position of the associate receiving trucks for shipping truck t^s_i.

b. In set T^p, move the product routing α_{pi} to the corresponding position of t^r_p above. Move the product routing α_{pj} to the position next to product routing α_{pi}.

c. In set T^n, move β_{pi} and β_{pj} to the corresponding position of α_{pi} and α_{pj} above, respectively.

The modified sequences in sets T^r, T^p and T^n will then appear as in sequence *(6-23)*.

$$
\begin{array}{lllll}
 & \overbrace{\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_h} & \overbrace{\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_i} & \overbrace{\qquad\qquad\qquad}^{\text{Associate Receiving Trucks for } t^s_j} & \\
T^r = \{\ldots, & \ldots, & \ldots, t^r_a, t^r_b, \ldots, t^r_p & \ldots, t^r_c, t^r_d, \ldots, & \ldots\}. \\
T^s = \{\ldots, & t^s_h, & t^s_i, & t^s_j, & \ldots\}. \\
T^p = \{\ldots, & \ldots, & \ldots, \alpha_{ai}, \alpha_{bi}, \ldots, \alpha_{pi}, & \alpha_{pj}, \ldots, \alpha_{cj}, \alpha_{dj}, \ldots, & \ldots\}. \\
T^n = \{\ldots, & \ldots, & \ldots, \beta_{ai}, \beta_{bi}, \ldots, \beta_{pi}, & \beta_{pj}, \ldots, \beta_{cj}, \beta_{dj}, \ldots, & \ldots\}.
\end{array}
\qquad (6\text{-}23)
$$

2. Shipping trucks t^s_i and t^s_j have in common more than one associate receiving trucks t^r_p, t^r_q, t^r_r and t^r_m. Assume that $\beta_{pj} \geq D > \beta_{qj} \geq \beta_{rj} \geq \beta_{mj}$ for explanation. The following notations are defined for explanation:

   $t^r_{\varepsilon'}$ = Receiving truck whose $\beta_{\varepsilon j}$, is the largest among the common associate receiving trucks (i.e. $t^r_{\varepsilon'} = t^r_p$),

   $t^r_{\varepsilon''}$ = Receiving truck whose $\beta_{\varepsilon j}$ is the second largest among the common associate receiving trucks (i.e. $t^r_{\varepsilon''} = t^r_q$).

   i) If both of the following two conditions, *C4* and *C5*, are satisfied; the sequences will appear as in sequence *(6-24)*.

      <u>C4</u>. Receiving truck $t^r_{\varepsilon'}$ (i.e. receiving truck t^r_p) is the first scheduled receiving truck among the associate receiving trucks of shipping truck t^s_i.

      <u>C5</u>. Receiving truck $t^r_{\varepsilon'}$ (i.e. receiving truck t^r_p) is the last scheduled receiving truck among the associate receiving trucks of shipping truck t^s_h.

Associate Receiving Trucks for t^s_h | Associate Receiving Trucks for t^s_i | Associate Receiving Trucks for t^s_j

$$T^r = \{\ldots,\ \ldots, t^r_p,\ \ t^r_p, t^r_q, \ldots, t^r_m, \ldots, t^r_r, \ldots,\ \ldots, t^r_a, t^r_r, t^r_q, t^r_p, \ldots, t^r_m, \ldots,\ \ldots\}.$$

$$T^s = \{\ldots,\ t^s_h,\ \ t^s_i,\ \ t^s_j,\ \ldots\}.$$

$$T^p = \{\ldots,\ \alpha_{ph},\ \ \alpha_{pi}, \alpha_{qi}, \ldots, \alpha_{mi}, \ldots, \alpha_{ri}, \ldots,\ \ldots, \alpha_{aj}, \alpha_{rj}, \alpha_{qj}, \alpha_{pj}, \ldots, \alpha_{mj}, \ldots,\ \ldots\}.$$

$$T^p = \{\ldots,\ \beta_{ph},\ \ \beta_{pi}, \beta_{qi}, \ldots, \beta_{mi}, \ldots, \beta_{ri} \ldots,\ \ldots, \beta_{aj}, \beta_{rj}, \beta_{qj}, \beta_{pj}, \ldots, \beta_{mj}, \ldots,\ \ldots\}. \quad (6\text{-}24)$$

$\beta_{rj}<D$ $\quad \beta_{qj}<D$ $\quad \beta_{pj}\geq D$ $\quad \beta_{mj}<D$

Then sequences in sets T^r, T^p and T^s can be modified as follows to decrease makespan:

   a. In set T^r, remove all associate receiving trucks $t^r_{\bar{\varepsilon}}$ of shipping truck t^s_j whose $\beta_{\bar{\varepsilon}j}$ is less than truck change time among the common associate receiving trucks; (i.e. remove the associate receiving trucks t^r_q, t^r_r and t^r_m of shipping truck t^s_j). Move the associate receiving truck $t^r_{\varepsilon''}$ (i.e. t^r_q) of shipping truck t^s_i to the last associate receiving truck position for shipping truck t^s_i.

b. In set T^p, move the product routing $\alpha_{\varepsilon^- i}$ (i.e. α_{qi}) to the corresponding position of $t^r_{\varepsilon^-}$ (i.e. t^r_q) above. Move all product routings α_{zj} whose β_{zj} is less than truck change time among the common associate receiving trucks (i.e. α_{qj}, α_{rj} and α_{mj}) to the position immediately following the product routings α_{zi} (i.e. α_{qi}, α_{ri} and α_{mi}), respectively.

c. In set T^n, move $\beta_{\varepsilon^- i}$ (i.e. β_{qi}) to the corresponding position of $\alpha_{\varepsilon^- i}$ (i.e. α_{qi}) above. Move all β_{zj}, which is less than truck change time among the common associate receiving trucks, (i.e. β_{qj}, β_{rj} and β_{mj}) to the corresponding position of product routings α_{zj} (i.e. α_{qj}, α_{rj} and α_{mj}) above, respectively.

The modified sequences in sets T^r, T^p and T^n will appear as in sequence *(6-25)*.

Associate Receiving Trucks for t^s_h | Associate Receiving Trucks for t^s_i | Associate Receiving Trucks for t^s_j

$$T^r = \{\ldots,\ \ldots, t^r_p,\ t^r_p, \ldots, t^r_m, \ldots, t^r_r, \ldots, t^r_q,\ \ldots t^r_a, t^r_p, \ldots,\ \ldots\}.$$
$$T^s = \{\ldots,\ t^s_h,\ t^s_i,\ t^s_j,\ \ldots\}.$$
$$T^p = \{\ldots,\ \alpha_{ph},\ \alpha_{pi}, \ldots, \alpha_{mi}, \alpha_{mj}, \ldots, \alpha_{ri}, \alpha_{rj}, \ldots, \alpha_{qi},\ \alpha_{qj}, \ldots, \alpha_{aj}, \alpha_{pj}, \ldots,\ \ldots\}.$$
$$T^p = \{\ldots,\ \beta_{ph},\ \beta_{pi}, \ldots, \beta_{mi}, \beta_{mj}, \ldots, \beta_{ri}\ \beta_{rj}, \ldots, \beta_{qi},\ \beta_{qj}, \ldots, \beta_{aj}, \beta_{pj}, \ldots,\ \ldots\}.$$

(6-25)

ii) Otherwise (i.e. if any of the above two conditions *C4* or *C5* is not satisfied), the sequences may appear as in sequence *(6-26)*.

Associate Receiving Trucks for t^s_h | Associate Receiving Trucks for t^s_i | Associate Receiving Trucks for t^s_j

$$T^r = \{\ldots,\ \ldots,\ t^r_r, t^r_p, \ldots, t^r_q, \ldots, t^r_m, \ldots,\ \ldots, t^r_a, t^r_m, t^r_p, t^r_q, \ldots, t^r_r, \ldots,\ \ldots\}.$$
$$T^s = \{\ldots,\ t^s_h,\ t^s_i,\ t^s_j,\ \ldots\}.$$
$$T^p = \{\ldots,\ \ldots,\ \alpha_{ri}, \alpha_{pi}, \ldots, \alpha_{qi}, \ldots, \alpha_{mi}, \ldots,\ \ldots, \alpha_{aj}, \alpha_{mj}, \alpha_{pj}, \alpha_{qj}, \ldots, \alpha_{rj}, \ldots,\ \ldots\}.$$
$$T^n = \{\ldots,\ \ldots,\ \beta_{ri}, \beta_{pi}, \ldots, \beta_{qi}, \ldots, \beta_{mi} \ldots,\ \ldots, \beta_{aj}, \beta_{mj}, \beta_{pj}, \beta_{qj}, \ldots, \beta_{rj}, \ldots,\ \ldots\}.$$

(6-26)

$\beta_{mj}<D$ $\quad\beta_{pj}\geq D$ $\quad\beta_{qj}<D$ $\quad\beta_{rj}<D$

Then sequences in sets T^r, T^p and T^n can be modified as follows:

a. In set T^r, remove all associate receiving trucks t^r_z of shipping truck t^s_j whose β_{zj} is less than truck change time among the common associate receiving trucks; (i.e. remove the associate receiving trucks t^r_q, t^r_r and t^r_m of shipping truck t^s_j). Remove the associate receiving truck $t^r_{\varepsilon'}$ (i.e. t^r_p) of shipping truck t^s_j. Move the associate receiving truck $t^r_{\varepsilon'}$ (i.e. t^r_p) of shipping truck t^s_i to the last associate receiving truck position for shipping truck t^s_i.

b. In set T^p, move the product routing $\alpha_{\varepsilon\bar{i}}$ (i.e. α_{pi}) to the corresponding position of $t^r_{\varepsilon'}$ (i.e. t^r_p) above. Move the product routing $\alpha_{\varepsilon\bar{j}}$ (i.e. α_{pj}) to the position next to product routing $\alpha_{\varepsilon\bar{i}}$ (i.e. α_{pi}). Move all product routings α_{zj} whose β_{zj} is less than truck change time among the common associate receiving trucks (i.e. α_{qj}, α_{rj} and α_{mj}) to the position next to product routings α_{zi} (i.e. α_{qi}, α_{ri} and α_{mi}), respectively.

c. In set T^n, move $\beta_{\varepsilon\bar{i}}$ and $\beta_{\varepsilon\bar{j}}$ (i.e. β_{pi} and β_{pj}) to the corresponding position of $\alpha_{\varepsilon\bar{i}}$ and $\alpha_{\varepsilon\bar{j}}$ (i.e. α_{pi} and α_{pj}) above, respectively. Move all β_{zj}, which are less than truck change time among the common associate receiving trucks, (i.e. β_{qj}, β_{rj} and β_{mj}) to the corresponding position of product routings α_{zj} (i.e. α_{qj}, α_{rj} and α_{mj}) above, respectively.

The modified sequences in sets T^r, T^p and T^n will appear as in sequence *(6-27)*.

Associate Receiving Trucks for t^s_h | Associate Receiving Trucks for t^s_i | Associate Receiving Trucks for t^s_j

$$
\begin{array}{llllllll}
T^r = \{\ldots, & \ldots, & t^r_r, \ldots, & t^r_q, \ldots, & t^r_m, \ldots, & t^r_p, & \ldots, t^r_a, \ldots, & \ldots\}. \\
T^s = \{\ldots, & t^s_h, & & t^s_i, & & & t^s_j, & \ldots\}. \\
T^p = \{\ldots, & \ldots, & \alpha_{ri}, \alpha_{rj}, \ldots, & \alpha_{qi}, \alpha_{qj}, \ldots, & \alpha_{mi}, \alpha_{mj}, \ldots, & \alpha_{pi}, & \alpha_{pj}, \ldots, \alpha_{aj}, \ldots, & \ldots\}. \\
T^n = \{\ldots, & \ldots, & \beta_{ri}, \beta_{rj}, \ldots, & \beta_{qi}, \beta_{qj}, \ldots, & \beta_{mi}, \beta_{mj}, \ldots, & \beta_{pi}, & \beta_{pj}, \ldots, \beta_{aj}, \ldots, & \ldots\}.
\end{array}
\qquad (6\text{-}27)
$$

Now, consider the second condition for *Phase II*. Because of the characteristics of the heuristic algorithm for the *Case 3* problem, each shipping truck appears only once in the shipping truck sequence. However, a receiving truck may need to come into the dock several times to unload its products. So a receiving truck may appear more than once in the receiving truck sequence. The second condition for *Phase II* checks whether makespan can be decreased by reducing the number of receiving truck changes. In order to reduce the number of receiving truck changes, receiving trucks that appear more than once in the receiving truck sequence are considered. By reducing the number of revisits of the same receiving truck into the dock, the number of receiving truck changes can be decreased. Decreasing the number of repeat visits into the dock by a receiving truck, of course, will cause the unloading of more products, which are not needed by the current shipping truck, and therefore these products will have to be sent to temporary storage. Unloading more products not needed by the current shipping truck may increase the waiting time of the shipping truck at the dock, which in turn may delay the departure time of the current shipping truck. However, makespan can be decreased by decreasing the delay time due to receiving truck changes.

Makespan can be decreased whenever the condition presented below is satisfied. For each scheduled receiving truck in set T^r, the total number of products unloaded, when it comes into the dock, can be found from the information in sets T^r, T^p and T^n. Suppose that the time required to unload products from receiving truck t^r_p in a certain position in the receiving truck sequence is less than the delay time for receiving truck change D. This means makespan can be decreased by unloading the products from receiving truck t^r_p and sending them to temporary storage rather than changing the receiving truck, if possible. Therefore, if the time required to unload products from receiving truck t^r_p is less than the delay time for truck change D, and it is not the first occurenence of receiving truck t^r_p in the receiving truck sequence, then makespan can be decreased by unloading the products at the earlier occurence of receiving truck t^r_p in the receiving truck sequence. The second condition for *Phase II* is explained as follows:

Condition 2

This condition only applies to receiving trucks that appear more than once in the receiving truck sequence. Suppose the time required to unload products from the associate receiving truck t^r_p of shipping truck t^s_j is less than the delay time for truck change D. Additionally, suppose that the receiving truck t^r_p is already scheduled earlier in the receiving truck sequence before it is scheduled as an associate receiving truck for shipping truck t^s_j. If the scheduling of the associate receiving truck t^r_p of shipping truck t^s_i is the last time t^r_p is scheduled before it (i.e., t^r_p) is scheduled again as an associate for shipping truck t^s_j, then the sequences in sets T^r, T^p and T^n are modified as follows:

1. If receiving truck t^r_p unloads products only for shipping truck t^s_j when it comes into dock, the sequences may be presented as in sequence *(6-28)*.

The last position of t^r_p scheduled prior to the position of t^r_p as an associate receiving truck for shipping truck t^s_j

Associate Receiving Trucks for t^s_i

Associate Receiving Trucks for t^s_k

Associate Receiving Trucks for t^s_j

Time required to unload products is less than the delay time for truck change

$$
\begin{aligned}
T^r &= \{\ldots, \quad \overbrace{, \ldots, t^r_a, t^r_p, t^r_b, \ldots,} \quad \overbrace{\ldots\ldots,} \quad \overbrace{\ldots, t^r_c, t^r_p, t^r_d, \ldots,} \quad \ldots\}.\\
T^s &= \{\ldots, \quad t^s_i, \quad t^s_k, \quad t^s_j, \quad \ldots\}.\\
T^p &= \{\ldots, \quad , \ldots, \alpha_{ai}, \alpha_{pi}, \alpha_{bi}, \ldots, \quad \ldots\ldots, \quad \ldots, \alpha_{cj}, \alpha_{pj}, \alpha_{dj}, \ldots, \quad \ldots\}.\\
T^n &= \{\ldots, \quad , \ldots, \beta_{ai}, \beta_{pi}, \beta_{bi}, \ldots, \quad \ldots\ldots, \quad \ldots, \beta_{cj}, \beta_{pj}, \beta_{dj}, \ldots, \quad \ldots\}.
\end{aligned} \qquad (6\text{-}28)
$$

$\beta_{pj} < D$

Only α_{pj} is unloaded from receiving truck t^r_p

If $\beta_{pj} < D$, sequences in sets T^r, T^p and T^n can be modified as follows to decrease makespan:

a. In set T^r, remove the associate receiving truck t^r_p of shipping truck t^s_j.
b. In set T^p, move the product routing α_{pj} to the next position that follows product routing α_{pi}.
c. In set T^n, move β_{pj} to the corresponding position of α_{pj} above.

The modified sequences in sets T^r, T^p and T^n will now appear as in sequence *(6-29)*.

	Associate Receiving Trucks for t^s_i	Associate Receiving Trucks for t^s_k	Associate Receiving Trucks for t^s_j	
$T^r = \{\ldots,$	$, \ldots, t^r_a, t^r_p, \quad t^r_b, \ldots,$	$\ldots\ldots,$	$\ldots, t^r_c, t^r_d, \ldots,$	$\ldots\}.$
$T^s = \{\ldots,$	$t^s_i,$	$t^s_k,$	$t^s_j,$	$\ldots\}.$
$T^p = \{\ldots,$	$, \ldots, \alpha_{ai}, \alpha_{pi}, \alpha_{pj}, \alpha_{bi}, \ldots,$	$\ldots\ldots,$	$\ldots, \alpha_{cj}, \alpha_{dj}, \ldots,$	$\ldots\}.$
$T^n = \{\ldots,$	$, \ldots, \beta_{ai}, \beta_{pi}, \beta_{pj}, \beta_{bi}, \ldots,$	$\ldots\ldots,$	$\ldots, \beta_{cj}, \beta_{dj}, \ldots,$	$\ldots\}.$

(6-29)

2. If receiving truck t^r_p unloads products for more than one shipping truck t^s_j when it comes into dock, the sequences may be presented as in sequence *(6-30)*.

The last position of t^r_p scheduled prior to the position of t^r_p as an associate receiving truck for shipping truck t^s_j

Time required to unload products is less than the delay time for truck change

	Associate Receiving Trucks for t^s_i	Associate Receiving Trucks for t^s_k	Associate Receiving Trucks for t^s_j	
$T^r = \{\ldots,$	$\ldots, t^r_a, t^r_p, t^r_b, \ldots,$	$\ldots\ldots,$	$\ldots, t^r_c, \quad t^r_p, \quad t^r_d, \ldots,$	$\ldots\}.$
$T^s = \{\ldots,$	$t^s_i,$	$t^s_k,$	$t^s_j,$	$\ldots\}.$
$T^p = \{\ldots,$	$\ldots, \alpha_{ai}, \alpha_{pi}, \alpha_{bi}, \ldots,$	$\ldots\ldots,$	$\ldots, \alpha_{cj}, \alpha_{pj}, \alpha_{pm}, \alpha_{pn}, \alpha_{dj}, \ldots,$	$\ldots\}.$
$T^n = \{\ldots,$	$\ldots, \beta_{ai}, \beta_{pi}, \beta_{bi}, \ldots,$	$\ldots\ldots,$	$\ldots, \beta_{cj}, \beta_{pj}, \beta_{pm}, \beta_{pn}, \beta_{dj}, \ldots,$	$\ldots\}.$

(6-30)

$(\beta_{pj}+\beta_{pm}+\beta_{pn}) < D$
α_{pj}, α_{pm} and α_{pn} are unloaded from receiving truck t^r_p continuously.

If $(\beta_{pj}+\beta_{pm}+\beta_{pn}) < D$, sequences in sets T^r, T^p and T^n can be modified as follows to decrease makespan:

a. In set T^r, remove the associate receiving truck t^r_p of shipping truck t^s_j.

b. In set T^p, move the product routings α_{pj}, α_{pm} and α_{pn} to the next positions that follow product routing α_{pi}.

c. In set T^n, move β_{pj}, β_{pm} and β_{pn} to the corresponding positions of α_{pj}, α_{pm} and α_{pn} above, respectively

The modified sequences in sets T^r, T^p and T^n will now appear as in sequence *(6-31)*.

$$
\begin{array}{lllll}
 & \overbrace{\ldots, t^r_a, t^r_p, \quad t^r_b, \ldots,}^{\text{Associate Receiving Trucks for } t^s_i} & \overbrace{\ldots\ldots,}^{\text{Associate Receiving Trucks for } t^s_k} & \overbrace{\ldots, t^r_c, t^r_d, \ldots,}^{\text{Associate Receiving Trucks for } t^s_j} & \\
T^r = \{\ldots, & \ldots, t^r_a, t^r_p, \quad t^r_b, \ldots, & \ldots\ldots, & \ldots, t^r_c, t^r_d, \ldots, & \ldots\}. \\
T^s = \{\ldots, & t^s_i, & t^s_k, & t^s_j, & \ldots\}. \\
T^p = \{\ldots, & \ldots, \alpha_{ai}, \alpha_{pi}, \alpha_{pj}, \alpha_{pm}, \alpha_{pn}, \alpha_{bi}, \ldots, & \ldots\ldots, & \ldots, \alpha_{cj}, \alpha_{dj}, \ldots, & \ldots\}. \\
T^n = \{\ldots, & \ldots, \beta_{ai}, \beta_{pi}, \beta_{pj}, \beta_{pm}, \beta_{pn}, \beta_{bi}, \ldots, & \ldots\ldots, & \ldots, \beta_{cj}, \beta_{dj}, \ldots, & \ldots\}.
\end{array}
\tag{6-31}
$$

The algorithm for *Phase* II will change the schedules if any of the above two conditions are satisfied. The procedure for *Phase II* is presented as follows:

<u>*PHASE II* OF THE HEURISTIC ALGORITHM FOR THE *CASE 3* PROBLEM</u>

<u>STEP 1</u>

From *Phase I*, the solution for the receiving truck and shipping truck sequences, and information about product routing and the number of products transferred from the scheduled receiving trucks to the scheduled shipping trucks are known. Take this information as input.

<u>STEP 2</u>

Starting from the beginning of the shipping truck sequence in set T^s, sequentially investigate two consecutive shipping trucks, t^s_i and t^s_j, and their associate receiving trucks.

- <u>*2a*</u> If any two consecutive shipping trucks in the shipping truck sequence, t^s_i and t^s_j, have the same associate receiving truck t^r_p, modify the sequences in sets T^r, T^p and T^n as presented in *Condition 1*.
- <u>*2b*</u> If all shipping trucks in set T^s are investigated, go to *Step 3*. Otherwise, go to *Step 2a* in order to investigate the next two consecutive shipping trucks.

STEP 3

Starting from the end of set T^r, check *Condition 2* presented above. If *Condition 2* is satisfied, modify the sequences in sets T^r, T^p and T^n as presented in *Condition 2*. Continue *Step 3* until the first scheduled receiving truck in set T^r is checked.

STEP 4

Stop. The best solution for the *Case 3* problem is found. Sets T^r and T^s show the receiving truck sequence and the shipping truck sequence, respectively. Product routing and the total number of products transferred between the scheduled receiving and shipping trucks are presented in sets T^p and T^n, respectively.

Figure 16 describes the algorithmic steps of the heuristic algorithm of *Phase II*. To illustrate *Phase II* algorithm for the *Case 3* problem, consider Example 4 problem that was presented earlier under *Phase I* in Section 6.2.2.2. In *Step 1* of *Phase II*, the following information is known from the solution of *Phase I* as presented in sequence *(6-19)*.

$$
\begin{array}{lccccc}
T^r = \{ & \overbrace{t^r_4,}^{\substack{\text{Associate Receiving}\\ \text{Trucks for } t^s_4}} & \overbrace{t^r_5, t^r_2,}^{\substack{\text{Associate Receiving}\\ \text{Trucks for } t^s_1}} & \overbrace{t^r_1,\ t^r_4,\ t^r_2,\ t^r_5,}^{\substack{\text{Associate Receiving}\\ \text{Trucks for } t^s_3}} & \overbrace{t^r_3,\ t^r_5,\ t^r_2,\ t^r_4, t^r_1}^{\substack{\text{Associate Receiving}\\ \text{Trucks for } t^s_2}} & \}. \\
T^s = \{ & t^s_4, & t^s_1, & t^s_3, & t^s_2 & \}. \\
T^p = \{ & \alpha_{44}, & \alpha_{51}, \alpha_{21}, & \alpha_{13}, \alpha_{43}, \alpha_{23}, \alpha_{53}, & \alpha_{32}, \alpha_{52}, \alpha_{22}, \alpha_{42}, \alpha_{12} & \}. \\
T^n = \{ & 80, & 131, 50, & 166, 115, 51,\ 4, & 190, 85,\ 59,\ 35,\ 14 & \}.
\end{array}
$$

In *Step 2*, the first two shipping trucks $\{t^s_4, t^s_1\}$ in set T^s and their associate receiving trucks $\{t^r_4, t^r_5, t^r_2\}$ in set T^r are investigated. Because all receiving trucks appeared only once in the sequence, the next two shipping trucks $\{t^s_1, t^s_3\}$ in set T^s, and their associate receiving trucks $\{t^r_5, t^r_2, t^r_1, t^r_4, t^r_2, t^r_5\}$ in set T^r are investigated. In this step, receiving trucks t^r_2 and t^r_5 are found to be scheduled twice for the consecutively scheduled shipping trucks t^s_1 and t^s_3. Following sequence *(6-27)* in *Condition 1*, sets T^r, T^p and T^n are scheduled. After the sets T^r, T^p and T^n are rescheduled, the new schedule now appears as follows:

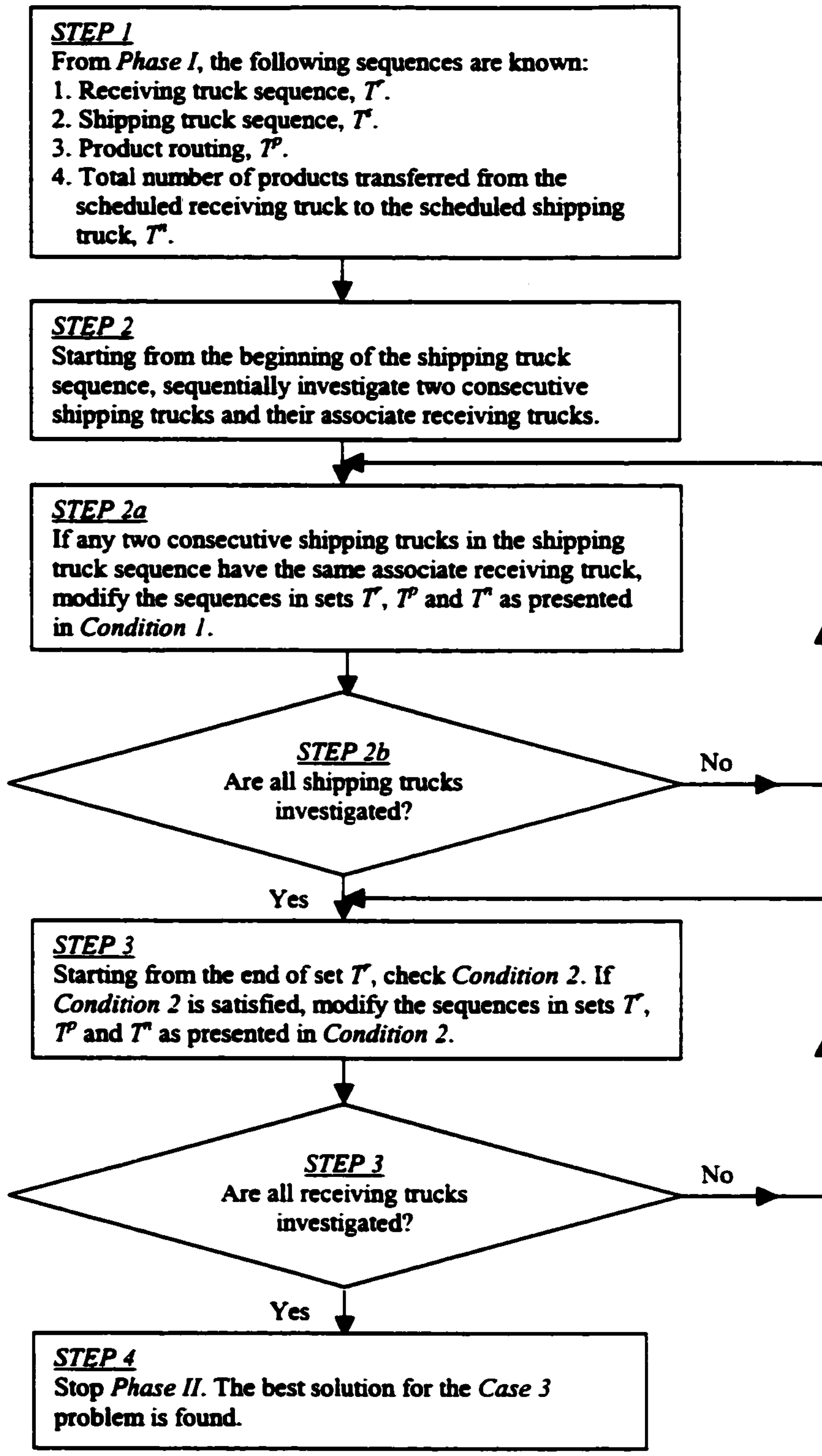


Figure 16. *Phase II* of the Heuristic Algorithm for the *Case 3* Problem

Associate Receiving Trucks for t^s_4 | Associate Receiving Trucks for t^s_1 | Associate Receiving Trucks for t^s_3 | Associate Receiving Trucks for t^s_2

$T^r = \{ \; t^r_4, \quad t^r_5, \; t^r_2, \quad t^r_1, \; t^r_4, \quad t^r_3, \; t^r_5, \; t^r_2, \; t^r_4, \; t^r_1 \}.$

$T^s = \{ \; t^s_4, \quad t^s_1, \quad t^s_3, \quad t^s_2 \}.$

$T^p = \{ \; \alpha_{44}, \quad \alpha_{51}, \alpha_{53}, \alpha_{21}, \quad \alpha_{23}, \alpha_{13}, \alpha_{43}, \quad \alpha_{32}, \alpha_{52}, \alpha_{22}, \alpha_{42}, \alpha_{12} \}.$

$T^n = \{ \; 80, \quad 131, \; 4, \; 50, \quad 51, \; 166, \; 115, \quad 190, \; 85, \; 59, \; 35, \; 14 \}.$

α_{51} and α_{53} are unloaded from receiving truck t^r_5 continuously.

α_{21} and α_{23} are unloaded from receiving truck t^r_2 continuously.

Note that receiving truck t^r_2 is placed at the last position of the associate receiving trucks for shipping truck t^s_1 because β_{23} (=51) is larger than β_{53} (=4).

Now, the next two shipping trucks $\{t^s_3, t^s_2\}$ in set T^s, and their associate receiving trucks $\{t^r_1, t^r_4, t^r_3, t^r_5, t^r_2, t^r_4, t^r_1\}$ in set T^r are investigated. Receiving trucks t^r_1 and t^r_4 are scheduled twice. Following sequence *(6-27)* in *Condition 1* again, sets T^r, T^p and T^n are rescheduled. After the sequences are modified, the new schedule at the end of *Step 2* is as shown below:

Associate Receiving Trucks for t^s_4 | Associate Receiving Trucks for t^s_1 | Associate Receiving Trucks for t^s_3 | Associate Receiving Trucks for t^s_2

$T^r = \{ \; t^r_4, \quad t^r_5, \; t^r_2, \quad t^r_1, \; t^r_4, \quad t^r_3, \; t^r_5, \; t^r_2 \}.$

$T^s = \{ \; t^s_4, \quad t^s_1, \quad t^s_3, \quad t^s_2 \}.$

$T^p = \{ \; \alpha_{44}, \quad \alpha_{51}, \alpha_{53}, \alpha_{21}, \quad \alpha_{23}, \alpha_{13}, \alpha_{12}, \alpha_{43}, \quad \alpha_{42}, \alpha_{32}, \alpha_{52}, \alpha_{22} \}.$

$T^n = \{ \; 80, \quad 131, \; 4, \; 50, \quad 51, \; 166, \; 14, \; 115, \quad 35, \; 190, \; 85, \; 59 \}.$

α_{13} and α_{12} are unloaded from receiving truck t^r_1 continuously.

α_{43} and α_{42} are unloaded from receiving truck t^r_4 continuously.

In *Step 3*, there is no schedule that satisfies *Condition 2* as presented in Table 27. Table 27 shows the time required to unload products from a scheduled receiving truck in set T^r, when it comes to dock. Therefore, *Step 4* is invoked without modifying the schedules in *Step 3*. In *Step 4*, the best solution is found as shown below.

	Associate Receiving Trucks for t^s_4	Associate Receiving Trucks for t^s_1	Associate Receiving Trucks for t^s_3	Associate Receiving Trucks for t^s_2
$T^r = \{$	t^r_4,	t^r_5, t^r_2,	t^r_1, t^r_4,	t^r_3, t^r_5, t^r_2 $\}$.
$T^s = \{$	t^s_4,	t^s_1,	t^s_3,	t^s_2 $\}$.
$T^p = \{$	α_{44},	α_{51}, α_{53}, α_{21},	α_{23}, α_{13}, α_{12}, α_{43},	α_{42}, α_{32}, α_{52}, $\alpha_{22}\}$.
$T^n = \{$	80,	131, 4, 50,	51, 166, 14, 115,	35, 190, 85, 59$\}$.

Figures 17 and 18 show the Gantt charts of Example 4 for *Phase I* and *Phase II* of the heuristic algorithm for the *Case 3* problem, respectively. Makespan after *Phase I* for this example is 1,200. It decreased to 1,130 after *Phase II*.

6.2.2.4 Makespan

From the solution of the *Case 3* problem, the following information is known: the receiving truck sequence, the shipping truck sequence, the product routing, and the number of products transferred corresponding to the product routing. Suppose that the solution is presented as follows, where *[i]* represents the i^{th} sequence position in a set rather than the number *i* itself. For the sets T^p and T^n, *[i]'* refers to the receiving truck involved in the i^{th} sequence position while *[i]"* refers to the shipping truck involved in the i^{th} sequence position.

Table 27. Time Required unloading Products from Each Scheduled Receiving Truck

Receiving Truck Sequences	t^r_4	t^r_5	t^r_2	t^r_1	t^r_4	t^r_3	t^r_5	t^r_2
Time Required unloading Products	80	135	101	180	150	190	85	59

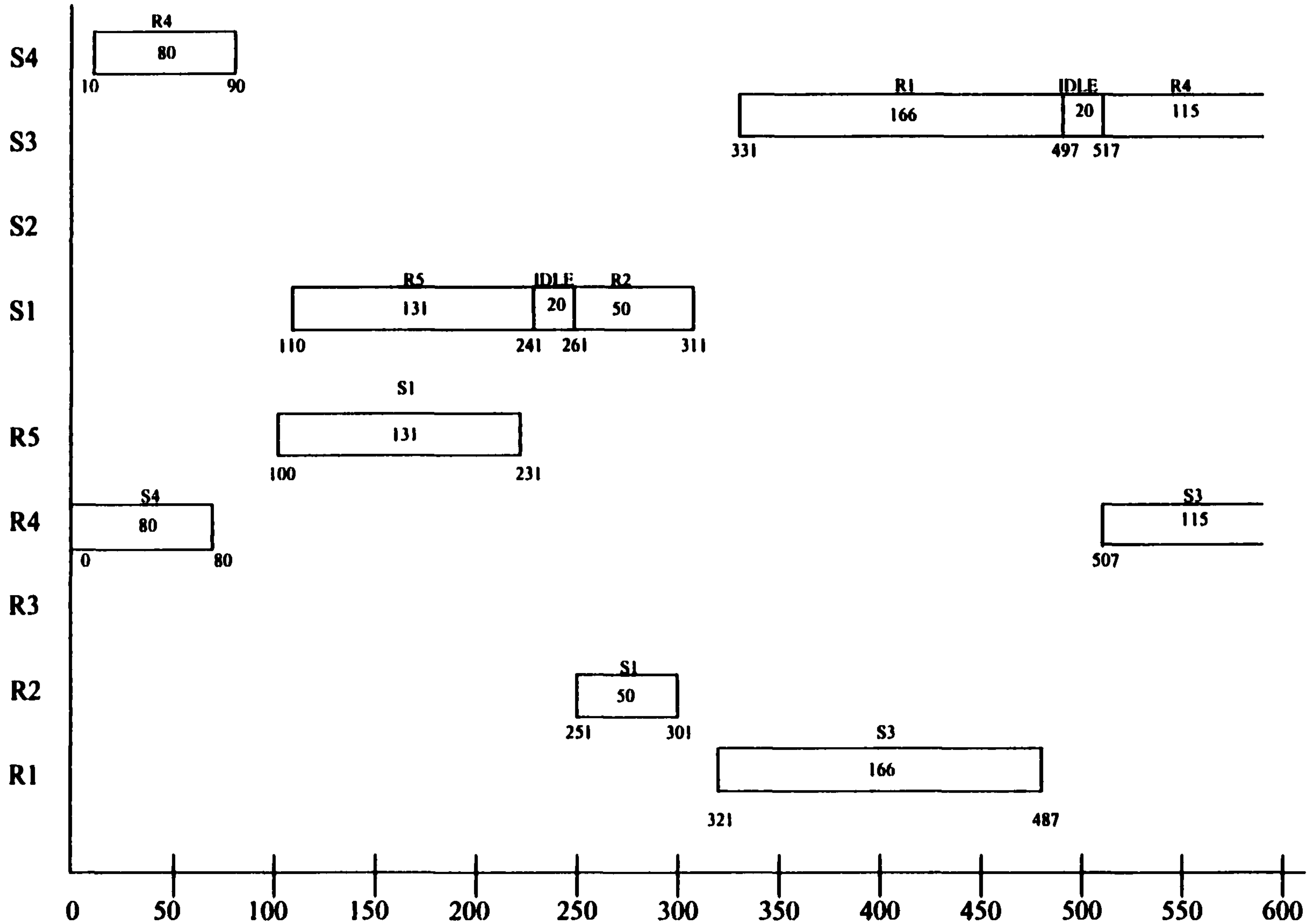


Figure 17. Gantt Chart of Example 4 after *Phase I* of the *Case 3* Problem

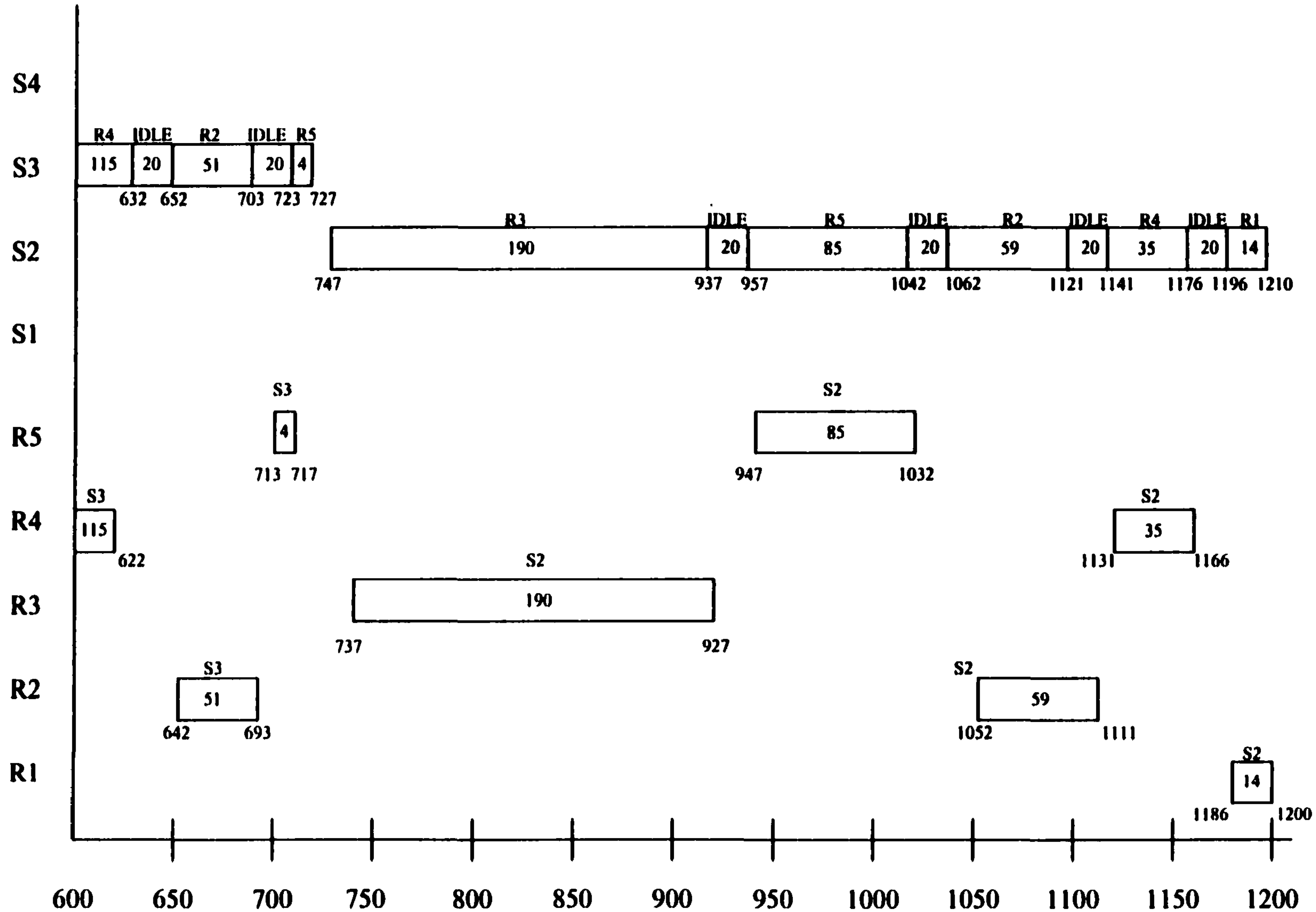


Figure 17. (continued)

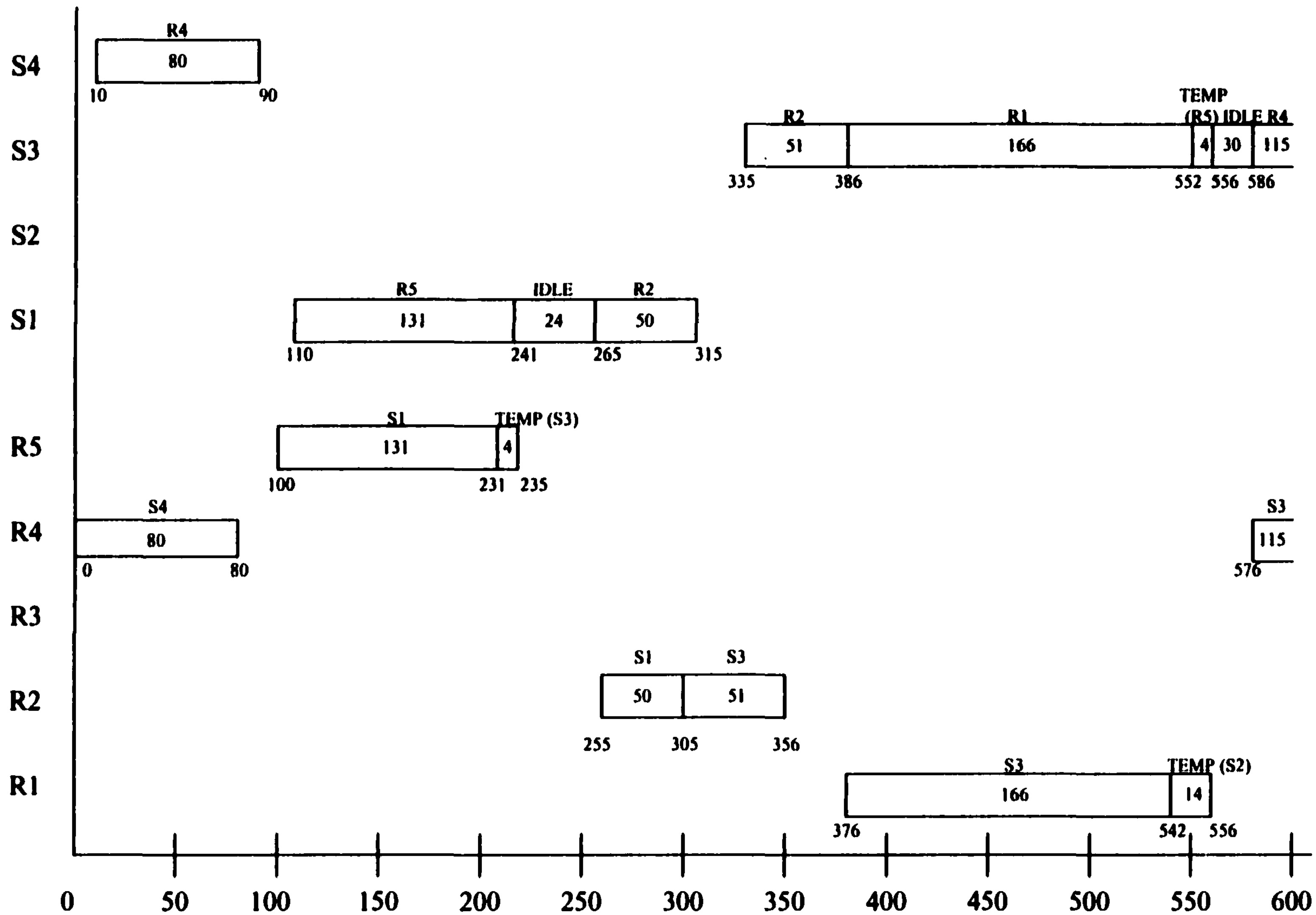


Figure 18. Gantt Chart of Example 4 after *Phase II* of the *Case 3* Problem

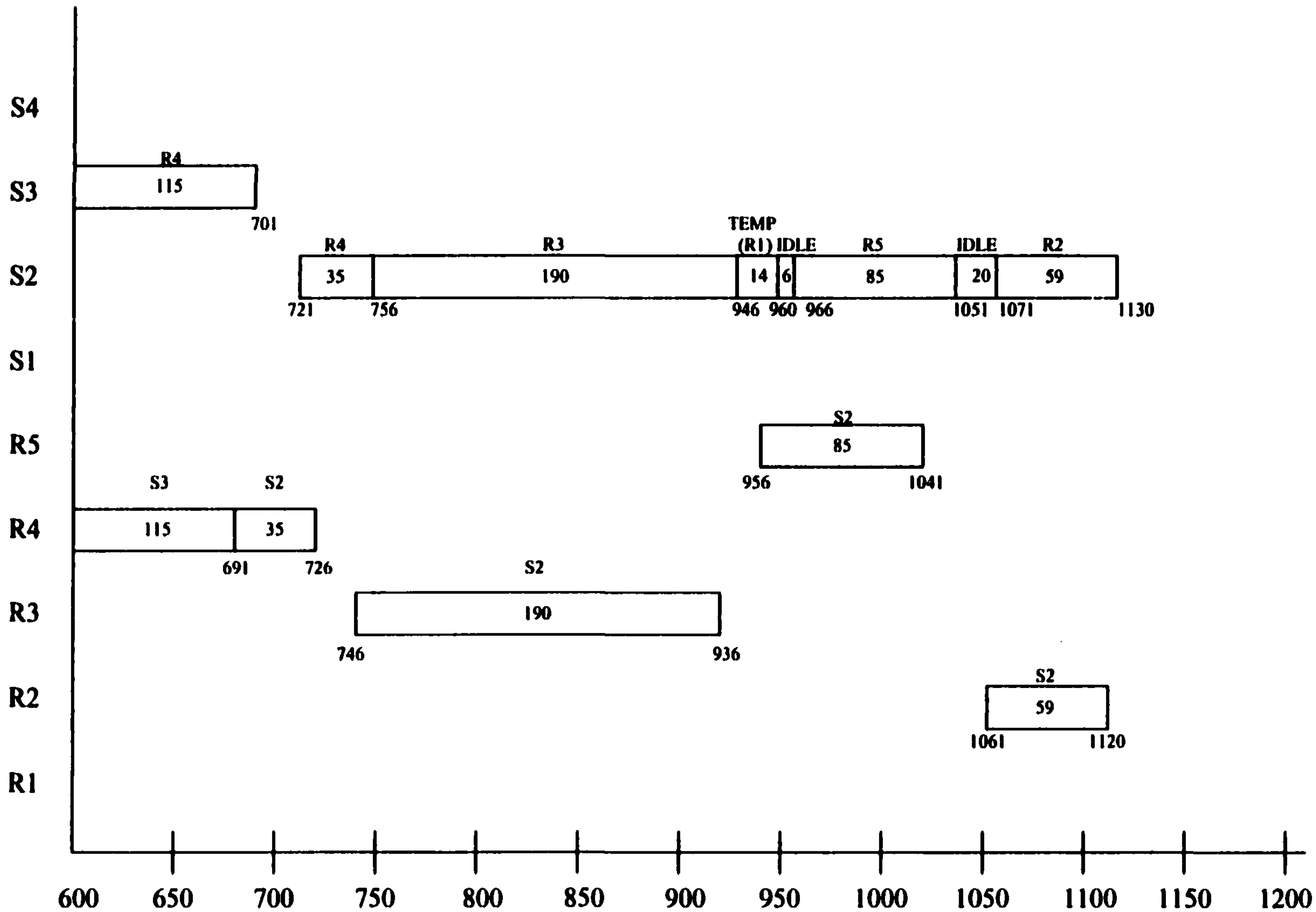


Figure 18. (continued)

$$T^r = \{ \quad t^r_{[1]},\ t^r_{[2]},\ \ldots,\ t^r_{[a-1]},\ t^r_{[a]} \quad \}.$$
$$T^s = \{ \quad t^s_{[1]},\ t^s_{[2]},\ \ldots,\ t^s_{[S-1]},\ t^s_{[S]} \quad \}.$$
$$T^p = \{ \ \alpha_{[1]'[1]''},\ \alpha_{[2]'[2]''},\ \alpha_{[3]'[3]''},\ \ldots,\ \alpha_{[b-1]'[b-1]''},\ \alpha_{[b]'[b]''} \ \}.$$
$$\mathrm{T}^n = \{ \ \beta_{[1]'[1]''},\ \beta_{[2]'[2]''},\ \beta_{[3]'[3]''},\ \ldots,\ \beta_{[b-1]'[b-1]''},\ \beta_{[b]'[b]''} \ \}.$$

Then makespan is calculated by finding two sets, F^r and F^s that contain task completion times. Elements in set F^r are the completion times of the activities associated with the product routing α_{ij} in set T^p. As defined previously, if γ_{ij} is the completion time of the receiving truck i corresponding to product routing α_{ij} in set T^p, set F^s is the ordered set of leaving or departure times of the shipping trucks corresponding to the shipping truck sequence in set T^s, and δ_i is the leaving or departure time of the shipping truck i corresponding to the shipping truck sequence in set T^s, then

$$F^r = \{ \ \gamma_{[1]'[1]''},\ \gamma_{[2]'[2]''},\ \gamma_{[3]'[3]''},\ \ldots,\ \gamma_{[b-1]'[b-1]''},\ \gamma_{[b]'[b]''} \ \}.$$
$$F^s = \{ \quad \delta_{[1]},\ \delta_{[2]},\ \ldots,\ \delta_{[S-1]},\ \delta_{[S]} \quad \}.$$

To calculate makespan, the procedure first finds the completion time $\gamma_{[i]'[i]''}$ in set F^r of the receiving truck $[i]'$ associated with the product routing $\alpha_{[i]'[i]''}$. Next, the departure time, $\delta_{[i]}$, in set F^s of the shipping truck $t^s_{[i]}$ in the shipping truck sequence is found. The procedure for calculating makespan is presented below.

1. Find the completion time of the receiving truck corresponding to each product routing in set T^p as follows (i.e. find $\gamma_{[i]'[i]''}$ in set F^r):

   i) For the first element, $\gamma_{[1]'[1]''}$, in set F^r.

   $$\gamma_{[1]'[1]''} = \beta_{[1]'[1]''}. \tag{6-32}$$

   The completion time of the first element, $\gamma_{[1]'[1]''}$, is simply the same as the unloading time of the first product routing $\alpha_{[1]'[1]''}$.

   ii) From the second element, $\gamma_{[2]'[2]''}$, to the last element, $\gamma_{[b]'[b]''}$, in set F^r.

   $$\gamma_{[i]'[i]''} = \gamma_{[i-1]'[i-1]''} + \beta_{[i]'[i]''}, \qquad \text{if } [i]' = [i-1]'. \tag{6-33-a}$$
   $$\gamma_{[i]'[i]''} = \gamma_{[i-1]'[i-1]''} + D + \beta_{[i]'[i]''}, \qquad \text{if } [i]' \neq [i-1]'. \tag{6-33-b}$$

The completion time from the second element to the last element is the same as the sum of the completion time of the previous element (i.e. $\gamma_{[i-1]\,\tau[i-1]''}$) and the unloading time of the corresponding product routing $\alpha_{[i]\,\tau[i]''}$ (i.e. $\beta_{[i]\,\tau[i]''}$) when there is no receiving truck change as presented in *equation (6-33-a)*. If there is a receiving truck change, the delay time for the receiving truck change from receiving truck $t^r_{[i-1]'}$ to receiving truck $t^r_{[i]'}$ must be added to the completion time, so *equation (6-33-b)* holds.

2. Find the departure times of the shipping trucks corresponding to the shipping truck sequence in set T^s (i.e. find $\delta_{[i]}$ in set F^s). In order to find the departure time of a shipping truck, the last position of the product routing associated with the shipping truck in set T^p must be identified first. Suppose that the last product routing associated with shipping truck $t^s_{[i]}$ is $\alpha_{[i^\bullet]\,\tau[i^\bullet]''}$, where $[i^\bullet]$ represents the last position of the product routing associated with shipping truck $t^s_{[i]}$. Then, $\delta_{[i]}$ in set F^s is calculated as follows:

   i) Find the first element, $\delta_{[1]}$, in set F^s corresponding to the first scheduled shipping truck $t^s_{[1]}$ as follows:

   $$\delta_{[1]} = \gamma_{[1^\bullet]\,\tau[1^\bullet]''} + V \tag{6-34}$$

   where the last product routing associated with shipping truck $t^s_{[1]}$ is $\alpha_{[1^\bullet]\,\tau[1^\bullet]''}$ in set T^p. *Equation (6-34)* says that the departure time of the first shipping truck is simply the same as the sum of the completion time of receiving truck $t^r_{[1^\bullet]'}$ corresponding to product routing $\alpha_{[1^\bullet]\,\tau[1^\bullet]''}$ in set T^p and the moving time of products from the receiving dock to the shipping dock.

   ii) From the second element, $\delta_{[2]}$, to the last element, $\delta_{[S]}$, in set F^s corresponding to shipping trucks $t^s_{[2]}$ through shipping truck $t^s_{[S]}$, find the departure times as follows:

   $$\delta_{[i]} = \max\left\{\gamma_{[i^\bullet]\,\tau[i^\bullet]''} + V,\ \delta_{[i-1]} + D + \sum_{k=1}^{N} s_{[i]k}\right\} \tag{6-35}$$

   again, where the last product routing associated with shipping truck $t^s_{[i]}$ is $\alpha_{[i^\bullet]\,\tau[i^\bullet]''}$ in set T^p.

   Equation (6-35) implies that the departure time from the second scheduled shipping truck to the last scheduled shipping truck is the larger value of two components,

$(\gamma_{[i^*]'[i^*]''} + V)$ and $(\delta_{[i-1]} + D + \sum_{k=1}^{N} s_{[i]k})$. The first component, $(\gamma_{[i^*]'[i^*]''} + V)$, is the sum of the completion time of receiving truck $t^r_{[i^*]'}$ corresponding to product routing $\alpha_{[i^*]'[i^*]''}$ in set T^p and the moving time of products from the receiving dock to the shipping dock. The second element, $(\delta_{[i-1]} + D + \sum_{k=1}^{N} s_{[i]k})$, is the sum of the departure time of shipping truck $t^s_{[i-1]}$, truck change time from shipping truck $t^s_{[i-1]}$ to shipping truck $t^s_{[i]}$ and the loading time of all needed products for shipping truck $t^s_{[i]}$.

3. After the departure times for all the shipping trucks in set F^s are found, makespan is $\delta_{[S]}$ which is the departure time of the last scheduled shipping truck or as given below in equation (6-36).

$$T = \delta_{[S]} \qquad (6\text{-}36\text{-a})$$

$$T = \max_{[i]} \{ \delta_{[i]} \} \qquad (6\text{-}36\text{-b})$$

To illustrate the procedure for calculating makespan, consider Example 4 presented in Section 6.2.2.2. The following solution, which is presented in sequence *(6-19)*, was found in Section 6.2.2.3. Assume that truck change time takes 20 units of time and moving time of products from the receiving dock to the shipping dock takes 10 units of time.

$T^r = \{\ t^r_4,\quad t^r_5,\ t^r_2,\quad t^r_1,\ t^r_4,\quad t^r_3,\ t^r_5,\ t^r_2\}.$

$T^s = \{\ t^s_4,\quad t^s_1,\quad t^s_3,\quad t^s_2\ \}.$

$T^p = \{\ \alpha_{44},\quad \alpha_{51}, \alpha_{53}, \alpha_{21}, \alpha_{23},\quad \alpha_{13}, \alpha_{12}, \alpha_{43}, \alpha_{42},\quad \alpha_{32}, \alpha_{52}, \alpha_{22}\}.$

$T^n = \{\ 80,\quad 131,\ 4,\ 50,\ 51,\quad 166, 14, 115, 35,\quad 190,\ 85,\ 59\}.$

The first step is to find the completion time, $\gamma_{[i]'[i]''}$, in set F^r of the receiving truck $[i]'$ associated with the product routing $\alpha_{[i]'[i]''}$ using the information in sets T^p and T^n. The first element, $\gamma_{[1]'[1]''} = \gamma_{44}$, is calculated as follow using equation (6-32):

$$\gamma_{[1]'[1]''}(=\gamma_{44}) = \beta_{[1]'[1]''}(=\beta_{44}) = 80.$$

In order to find the second element, $\gamma_{[2]'[2]''} = \gamma_{51}$, first investigate whether there is a receiving truck change for the consecutive elements in product routing. In this example, there is a receiving truck change because receiving truck $t^r_{[1]^r}$ corresponding to product routing $\alpha_{[1]^r[1]^{r*}}$ ($=\alpha_{44}$) is receiving truck t^r_4 and receiving truck $t^r_{[2]^r}$ corresponding to product routing $\alpha_{[2]^r[2]^{r*}}$ ($=\alpha_{51}$) is receiving truck t^r_5. Therefore, $\gamma_{[2]^r[2]^{r*}}$ ($=\gamma_{51}$) is calculated as follows using *equation (6-33-b)*:

$$\gamma_{[2]^r[2]^{r*}}(=\gamma_{51}) = \gamma_{[1]^r[1]^{r*}} + D + \beta_{[2]^r[2]^{r*}} = \gamma_{44} + D + \beta_{51} = 80 + 20 + 131 = 231.$$

For the third element, $\gamma_{[3]^r[3]^{r*}} = \gamma_{53}$, receiving truck t^r_5 stays in the dock because receiving truck $t^r_{[2]^r}$ corresponding to product routing $\alpha_{[2]^r[2]^{r*}}$ ($=\alpha_{51}$) and receiving truck $t^r_{[3]^r}$ corresponding to product routing $\alpha_{[3]^r[3]^{r*}}$ ($=\alpha_{53}$) are the same as receiving truck t^r_5. Therefore, $\gamma_{[3]^r[3]^{r*}}$ ($=\gamma_{53}$) is calculated as follows using *equation (6-33-a)*:

$$\gamma_{[3]^r[3]^{r*}}(=\gamma_{53}) = \gamma_{[2]^r[2]^{r*}} + \beta_{[3]^r[3]^{r*}} = \gamma_{51} + \beta_{53} = 231 + 4 = 235.$$

The procedure is repeated until all $\gamma_{[i]^r[i]^{r*}}$ in set F^r are found. The complete solution for set F^r is as presented below.

$$\begin{aligned} F^r &= \{\ \gamma_{[1]^r[1]^{r*}},\ \gamma_{[2]^r[2]^{r*}},\ \gamma_{[3]^r[3]^{r*}},\ \ldots,\ \gamma_{[11]^r[11]^{r*}},\ \gamma_{[12]^r[12]^{r*}}\ \} \\ &= \{\ \gamma_{44},\ \gamma_{51},\ \gamma_{53},\ \gamma_{21},\ \gamma_{23},\ \gamma_{13},\ \gamma_{12},\ \gamma_{43},\ \gamma_{42},\ \gamma_{32},\ \gamma_{52},\ \gamma_{22}\ \} \\ &= \{\ 80,\ 231, 235, 305, 356,\ 542, 556, 691, 726,\ 936, 1041,\ 1120\ \}. \end{aligned}$$

Next, the departure times of the scheduled shipping trucks are calculated. For the first element, $\delta_{[1]}$ ($=\delta_4$) in set F^s, the last product routing associated with shipping truck $t^s_{[1]}$ ($=t^s_4$) is $\alpha_{[1^*]^r[1^*]^{r*}} = \alpha_{[1]^r[1]^{r*}} = \alpha_{44}$. Then $\delta_{[1]}$ ($=\delta_4$) is calculated as follows using *equation (6-34)*:

$$\delta_{[1]}\ (=\delta_4) = \gamma_{[1^*]^r[1^*]^{r*}} + V = \gamma_{[1]^r[1]^{r*}} + V = \gamma_{44} + V = 80 + 10 = 90$$

To calculate the departure time for the second scheduled shipping truck, which in this example is shipping truck $t^s_{[2]}$ ($=t^s_1$), again the last product routing associated with shipping truck $t^s_{[2]}$ ($=t^s_1$) must be identified first. The last product routing associated with shipping truck $t^s_{[2]}$ ($=t^s_1$) is $\alpha_{[2^*]^r[2^*]^{r*}}$ ($=\alpha_{[4]^r[4]^{r*}} = \alpha_{21}$). Then $\delta_{[2]}$ ($=\delta_1$) is calculated as follows using *equation (6-35)*:

$$\delta_{[2]}(=\delta_1) = \max\left\{\gamma_{[2^*]^r[2^*]^{r*}} + V,\ \delta_{[1]} + D + \sum_{k=1}^{N} s_{[2]k}\right\} = \max\left\{\gamma_{[4]^r[4]^{r*}} + V,\ \delta_{[1]} + D + \sum_{k=1}^{N} s_{[2]k}\right\}$$

$$= \max\left\{\gamma_{21} + V,\ \delta_4 + D + \sum_{k=1}^{N} s_{1k}\right\} = \{305 + 10,\ 90 + 20 + 181\} = 315.$$

Note that the total number of products needed by shipping truck $t^s_{[2]}$ ($=t^s_1$) is 181. The same procedure is repeated to find $\delta_{[3]}$ ($=\delta_3$) and $\delta_{[4]}$ ($=\delta_2$) after the last product routing associated with shipping truck $t^s_{[3]}$ ($=t^s_3$) and $t^s_{[4]}$ ($=t^s_2$) are identified as $\alpha_{[3\bullet]'[3\bullet]''} = \alpha_{[8]'[8]''} = \alpha_{43}$ and $\alpha_{[4\bullet]'[4\bullet]''} = \alpha_{[12]'[12]''} = \alpha_{22}$, respectively. The complete solution for set F^s is presented below.

$$
\begin{array}{lllll}
F^s = \{ & \delta_{[1]}, & \delta_{[2]}, & \delta_{[3]}, & \delta_{[4]} \; \} \\
\quad = \{ & \delta_4, & \delta_1, & \delta_3, & \delta_2 \; \} \\
\quad = \{ & 90, & 315, & 701, & 1130 \; \}.
\end{array}
$$

Makespan for this problem is 1,130 ($=\delta_{[4]} = \delta_2$) which is the same as the solution found with Gantt Chart in Section 6.2.2.3.

6.3 Implementation and Results

The same twenty sets of problems as in *Cases 1* and *2* are used in applying and testing the heuristic algorithm for the *Case 3* problem. The optimal solutions for the problem instances could not be found using mathematical programming because of the large computational time required. For example, for *Test Problem Set 3*, which is the smallest problem among the twenty problems sets, with three receiving trucks, three shipping trucks and eight product types, attempt was made to solve it using LINDO and CPLEX. The mathematical model for the problem (i.e., *Test Problem Set 3*) has 307 decision variables that include 207 binary variables, 81 integer variables and 19 continuous variables. The number of constraints for the *Test Problem Set 3* is 304 consisting of 189 inequality constraints and 115 of equality constraints. Both softwares, LINDO and CPLEX, ran for more than a day and still could not find the optimal solution. (LINDO was implemented on a personal computer (Intel Pentium III Microprocessor 800MHz).) Moreover the convergence rate was very low. With the experience gained from *Test Problem Set 3*, it was evident that mathematical model approach is not practical to solve the *Case 3* problem for even small problems.

To assess the performance of the heuristic algorithm for the *Case 3* problem, the heuristic solutions needs to be compared with the optimal solutions for the *Case 3* problem instances that cannot be found using the mathematical programming model. However, the optimal solutions for the *Cases 1* and *2* models can be used as the upper bounds for the optimal solutions for the *Case 3* problem to assess the performance of the heuristic algorithm. The optimal solution for the *Case 3* problem must be, at least, as good as or better than the optimal solutions obtained for the *Cases 1* and *2* models since the *Case 3* problem is the most relaxed problem among the three cases studied in this research.

To test the heuristic algorithm for the *Case 3* problem, two different truck change times were used since the optimal solution is affected by the truck change time. For the first test, it is assumed that it takes 75 units of time to execute a truck change operation. Table 28 presents the optimal solutions for the *Cases 1* and *2* problems, and the heuristic solutions for the *Case 3* problem after applying the three different selection strategies in *Phase I*. The detailed heuristic solutions for the *Case 3* problem such as the receiving truck sequence, the shiping truck sequence and the product routing are presented in Appendix F. The second test is exactly the same as the first test except that 15 units of time is required to execute a truck change operation. The solutions for the second test are presented in Table 29. The detailed heuristic solutions of the second test for the *Case 3* problem are presented in Appendix G.

When the optimal solutions for the *Cases 1* and *2* problems are compared to each other based on the first test as shown in Table 28, the solution for the *Case 1* problem dominates that of the *Case 2* problem as expected. This is expected because the truck change time of 75 time units is relatively larger than the average length of time required to unload a batch of products from a receiving truck for the first test. (i.e. Unloading time of one time unit, which is the time it takes to unload one unit of product from a receiving truck, is much smaller than the truck change time of 75 time units in the first test.) Therefore, it is preferable to route the products through the temporary storage rather than incur delay due to truck change. On the other hand, the solution for the *Case 2* problem dominates that for the *Case 1* problem in the second test as presented in Table 29 because the truck change time of 15 time units is relatively smaller than the average length of time required to unload a batch of products from a receiving truck for the second test. In this case, it is preferable to incur the delay due to truck change rather than route the products through the temporary storage.

For the first test results presented in Table 28, the optimal solutions obtained from the *Case 1* model are better than those of the *Case 2* model in all twenty test problems. The compound heuristic solution of the three strategies for the *Case 3* problem found better solution than the optimal solution of the *Case 1* problem in thirteen test problems. The

Table 28. Makespans of the Optimal Solutions for the *Cases 1* and *2* Problems and the Heuristic Solutions for the *Case 3* Problem where Truck Change Time is 75

Problem Number	Case 1 Optimal	Case 2 Optimal	Case 3 Heuristics			
			Strategy 1	Strategy 2	Strategy 3	Compound
1	1557	1840	1509	1532	1480	1480
2	1577	1880	1683	1580	1580	1580
3	1372	1515	1354	1354	1354	1354
4	1749	2225	1940	1860	1912	1860
5	1579	1810	1484	1484	1484	1484
6	1546	1870	1497	1497	1495	1495
7	1535	1830	1549	1510	1510	1510
8	1525	1815	1461	1461	1451	1451
9	1473	1825	1443	1415	1440	1415
10	1452	1705	1399	1399	1399	1399
11	2232	2470	2320	2320	2263	2263
12	2833	3100	2800	2725	2725	2725
13	2386	2910	2330	2405	2526	2330
14	2385	2830	2392	2334	2380	2334
15	2745	3030	2802	2906	2745	2745
16	2407	2915	2574	2540	2465	2465
17	1867	2030	1805	1805	1730	1730
18	2502	2995	2620	2620	2695	2620
19	2553	2945	2495	2495	2495	2495
20	2732	3395	2938	3066	2863	2863

compound heuristic solution of the *Case 3* problem and the optimal soluton of the *Case 1* problem are the same in one test problem (*Test Set 15*). In six test problems, the optimal solution of the *Case 1* problem are better than the compound heuristic solution of the *Case 3* problem, but the differences are very small. The compound heuristic solution of the *Case 3* problem found better solution than the optimal solution for the *Case 2* problem in all twenty problems.

Table 29. Makespans of the Optimal Solutions for the *Cases 1* and *2* Problems and the Heuristic Solutions for the *Case 3* Problem where Truck Change Time is 15

Problem Number	Case 1 Optimal	Case 2 Optimal	Case 3 Heuristics			
			Strategy 1	Strategy 2	Strategy 3	Compound
1	1257	1240	1195	1195	1195	1195
2	1321	1280	1250	1250	1235	1235
3	1192	1095	1065	1065	1065	1065
4	1389	1325	1295	1265	1295	1265
5	1279	1210	1180	1180	1180	1180
6	1306	1270	1225	1225	1210	1210
7	1251	1230	1200	1200	1200	1200
8	1225	1155	1095	1095	1095	1095
9	1232	1165	1105	1105	1105	1105
10	1212	1165	1120	1120	1120	1120
11	1932	1870	1840	1840	1855	1840
12	2473	2260	2245	2230	2215	2215
13	2026	1950	1875	1875	1890	1875
14	2025	1990	1945	1960	1930	1930
15	2385	2310	2302	2295	2280	2280
16	2047	2015	1985	1985	1970	1970
17	1585	1430	1385	1385	1385	1385
18	2142	2095	2065	2035	2050	2035
19	2253	2045	1985	1985	1970	1970
20	2432	2375	2360	2345	2330	2330

Among the three heuristic solutions derived from the three strategies for the *Case 3* problem, the third strategy (i.e. maximum fitness) found the best solution in fifteen problems. The first strategy (i.e. maximum flow) found the best solution in six problems and the second strategy (i.e. maximum ratio) found it in eleven problems.

For the second test presented in Table 29, the optimal solutions of the *Case 2* model are better than those of the *Case 1* model in all twenty test problems. The compound heuristic solution obtained for the *Case 3* problem was better than the optimal solutions of the *Cases 1* and 2 models in all twenty test problems. Among the three heuristic solutions for the *Case 3* problem, the third strategy (i.e. maximum fitness) found the best solution in sixteen test problems. The first strategy (i.e. maximum flow) found the best solution in ten problems and the second strategy (i.e. maximum ratio) found it in twelve problems.

For further analysis of the performance of the heuristic algorithm, the percentage deviation of makespan between the optimal solutions of the *Cases 1* and *2* problems and the heuristic solutions for the *Case 3* problem were calculated, respectively. Percentage deviation is calculated as given below in *equation (6-37)*.

$$\begin{pmatrix} \textit{Percentage Deviation} \\ \textit{of Makespan (\%)} \end{pmatrix} = \frac{\begin{pmatrix} \textit{Makespan for Case3} \\ \textit{Heuristic Solutions} \end{pmatrix} - \begin{pmatrix} \textit{Makespan for Case k} \\ \textit{Optimal Solution} \end{pmatrix}}{\textit{Makespan for Case k Optimal Solution}} \times 100 \quad (6\text{-}37)$$

where $k = 1, 2$.

Table 30 presents the percentage deviation of makespan when 75 units of time are assumed for truck change time. As can be seen in Table 30, the average of percentage deviation between the optimal solutions for the *Case 1* problem and the compound heuristic solution for the *Case 3* problem is –1.39% for the twenty test problems. It means that the compound heuristic solutions for the *Case 3* problem is better than the optimal solutions for the *Case 1* problem on the average. The average percentage deviations between *strategies 1, 2* and *3* for the *Case 3* problem and the optimal solution for the *Case 1* model are 0.75%, 0.13% and –0.49%, respectively. It shows that every strategy for the *Case 3* problem found solutions which are very close to the optimal solutions for the *Case 1* model.

The percentage deviation of makespan when 15 units of the truck change time are assumed is shown in Table 31. The average percentage deviations for the twenty sets between the optimal solutions of the *Case 2* model and the compound heuristic solution of the *Case 3* problem is –3.19%. In this test, all three strategies for the *Case 3* problem found better solutions than the optimal solutions for the *Case 2* model on the average.

Among the three strategies of the heuristic algorithm for the *Case 3* problem, the average of the percentage deviation of *strategy 3* (maximum fitness) are the lowest in all cases regardless of truck change time. Even though *strategy 3* performed the best among the three strategies, it must be pointed out that the percentage deviation of *strategy 3* goes up to

Table 30. Percentage Deviation for Makespan between Optimal Solution for the *Case 1* Model and Heuristic Solutions for the *Case 3* Problem and between Optimal Solution for the *Case 2* Model and Heuristic Solutions for the *Case 3* Problem where Truck Change Time is 75

Problem Number	Between Optimal Solution for *Case 1* and Heuristics for *Case 3*				Between Optimal Solution for *Case 2* and Heuristics for *Case 3*			
	Strategy 1	Strategy 2	Strategy 3	Compound	Strategy 1	Strategy 2	Strategy 3	Compound
1	-3.08%	-1.61%	-4.95%	-4.95%	-17.99%	-16.74%	-19.57%	-19.57%
2	6.72%	0.19%	0.19%	0.19%	-10.48%	-15.96%	-15.96%	-15.96%
3	-1.31%	-1.31%	-1.31%	-1.31%	-10.63%	-10.63%	-10.63%	-10.63%
4	10.92%	6.35%	9.32%	6.35%	-12.81%	-16.40%	-14.07%	-16.40%
5	-6.02%	-6.02%	-6.02%	-6.02%	-18.01%	-18.01%	-18.01%	-18.01%
6	-3.17%	-3.17%	-3.30%	-3.30%	-19.95%	-19.95%	-20.05%	-20.05%
7	0.91%	-1.63%	-1.63%	-1.63%	-15.36%	-17.49%	-17.49%	-17.49%
8	-4.20%	-4.20%	-4.85%	-4.85%	-19.50%	-19.50%	-20.06%	-20.06%
9	-2.04%	-3.94%	-2.24%	-3.94%	-20.93%	-22.47%	-21.10%	-22.47%
10	-3.65%	-3.65%	-3.65%	-3.65%	-17.95%	-17.95%	-17.95%	-17.95%
11	3.94%	3.94%	1.39%	1.39%	-6.07%	-6.07%	-8.38%	-8.38%
12	-1.16%	-3.81%	-3.81%	-3.81%	-9.68%	-12.10%	-12.10%	-12.10%
13	-2.35%	0.80%	5.87%	-2.35%	-19.93%	-17.35%	-13.20%	-19.93%
14	0.29%	-2.14%	-0.21%	-2.14%	-15.48%	-17.53%	-15.90%	-17.53%
15	2.08%	5.87%	0.00%	0.00%	-7.52%	-4.09%	-9.41%	-9.41%
16	6.94%	5.53%	2.41%	2.41%	-11.70%	-12.86%	-15.44%	-15.44%
17	-3.32%	-3.32%	-7.34%	-7.34%	-11.08%	-11.08%	-14.78%	-14.78%
18	4.72%	4.72%	7.71%	4.72%	-12.52%	-12.52%	-10.02%	-12.52%
19	-2.27%	-2.27%	-2.27%	-2.27%	-15.28%	-15.28%	-15.28%	-15.28%
20	7.54%	12.23%	4.80%	4.80%	-13.46%	-9.69%	-15.67%	-15.67%
Average	0.57%	0.13%	-0.49%	-1.39%	-14.32%	-14.68%	-15.25%	-15.98%

Table 31. Percentage Deviation for Makespan between Optimal Solution for the *Case 1* Model and Heuristic Solutions for the *Case 3* Problem and between Optimal Solution for the *Case 2* Model and Heuristic Solutions for the *Case 3* Problem where Truck Change Time is 15

Problem Number	Between Optimal Solution for *Case 1* and Heuristics for *Case 3*				Between Optimal Solution for *Case 2* and Heuristics for *Case 3*			
	Strategy 1	Strategy 2	Strategy 3	Compound	Strategy 1	Strategy 2	Strategy 3	Compound
1	-4.93%	-4.93%	-4.93%	-4.93%	-3.63%	-3.63%	-3.63%	-3.63%
2	-5.37%	-5.37%	-6.51%	-6.51%	-2.34%	-2.34%	-3.52%	-3.52%
3	-10.65%	-10.65%	-10.65%	-10.65%	-2.74%	-2.74%	-2.74%	-2.74%
4	-6.77%	-8.93%	-6.77%	-8.93%	-2.26%	-4.53%	-2.26%	-4.53%
5	-7.74%	-7.74%	-7.74%	-7.74%	-2.48%	-2.48%	-2.48%	-2.48%
6	-6.20%	-6.20%	-7.35%	-7.35%	-3.54%	-3.54%	-4.72%	-4.72%
7	-4.08%	-4.08%	-4.08%	-4.08%	-2.44%	-2.44%	-2.44%	-2.44%
8	-10.61%	-10.61%	-10.61%	-10.61%	-5.19%	-5.19%	-5.19%	-5.19%
9	-10.31%	-10.31%	-10.31%	-10.31%	-5.15%	-5.15%	-5.15%	-5.15%
10	-7.59%	-7.59%	-7.59%	-7.59%	-3.86%	-3.86%	-3.86%	-3.86%
11	-4.76%	-4.76%	-3.99%	-4.76%	-1.60%	-1.60%	-0.80%	-1.60%
12	-9.22%	-9.83%	-10.43%	-10.43%	-0.66%	-1.33%	-1.99%	-1.99%
13	-7.45%	-7.45%	-6.71%	-7.45%	-3.85%	-3.85%	-3.08%	-3.85%
14	-3.95%	-3.21%	-4.69%	-4.69%	-2.26%	-1.51%	-3.02%	-3.02%
15	-3.48%	-3.77%	-4.40%	-4.40%	-0.35%	-0.65%	-1.30%	-1.30%
16	-3.03%	-3.03%	-3.76%	-3.76%	-1.49%	-1.49%	-2.23%	-2.23%
17	-12.62%	-12.62%	-12.62%	-12.62%	-3.15%	-3.15%	-3.15%	-3.15%
18	-3.59%	-5.00%	-4.30%	-5.00%	-1.43%	-2.86%	-2.15%	-2.86%
19	-11.90%	-11.90%	-12.56%	-12.56%	-2.93%	-2.93%	-3.67%	-3.67%
20	-2.96%	-3.58%	-4.19%	-4.19%	-0.63%	-1.26%	-1.89%	-1.89%
Average	-6.86%	-7.08%	-7.21%	-7.43%	-2.60%	-2.83%	-2.96%	-3.19%

9.32% in the worst case. To eliminate such poor solutions and ensure finding good solutions, the application of all three strategies is suggested. This can be done by implementing the compound heuristic algorithm for the *Case 3* problem that would improve the overall performance of the heuristic.

6.4 Conclusions

A mathematical model and a heuristic algorithm were developed to solve the cross docking problem for the *Case 3* problem. In the *Case 3* problem, trucks can come in and out of the dock multiple times until all their loads are unloaded or loaded depending whether it is a receiving truck or shipping truck. The decision to allow a truck to stay at the dock or leave the dock and be rescheduled later to continue its unloading or loading operation is based on what strategy can best reduce the overall makespan of the cross docking operation. Although the problem of scheduling the trucks at the docks to minimize makespan is clearly an optimization problem that can be modeled mathematically as done in this research, the computational requirement of such model is so prohibitive to render mathematical optimization ineffective as a solution approach. For such a mathematical model, the number of variables and constraints grows exponentially based on the number of receiving trucks, the number of shipping trucks, and the number of product types. Given the inability to establish the optimal solution for the test problems using mathematical programming, an alternate approach was adopted instead. The adopted approach employs the optimal solutions obtained for the *Cases 1* and *2* problems as the upper bounds for the solutions of the *Case 3* problem. This way, the performance of the heuristic algorithm developed could be benchmarked.

The heuristic algorithm consists of two phases. In *Phase I*, the initial receiving truck sequence, the initial shipping truck sequence and the product routing are determined. In the *Phase I* schedule, no products are sent to temporary storage. In order to find the associate receiving trucks in *Phase I*, three strategies were developed. The associate receiving trucks for a shipping truck is a set of candidate receiving trucks that can fully satisfy from their consignments the needs of the shipping truck. In *Phase II*, the receiving truck sequence and the product routing are modified as necessary to reduce makespan by sending some products to the temporary storage instead of changing trucks at the dock. The execution of *Phase II* is continued until makespan can no longer be decreased any further by modifying the current schedules.

The run time of the heuristic algorithm is very short to produce a solution. Because the solutions for the *Case 3* problem depends on the delay time for truck change (i.e., D) and the unloading time (i.e. u_k) for one unit of product type k, two different truck change times were used to test the performance of the heuristic algorithm. In the first twenty instances of test problems, 75 units of time were used for a truck change operation. Because D is relatively larger than the average length of time required to unload a batch of products from a receiving truck in this case, the optimal solutions for the *Case 1* model are better than those of the *Case 2* model in all twenty problem instances. In the second set of twenty test problems, 15 units of truck change time were used. In this case, the optimal solutions for the *Case 2* model are better than those of *Case 1* model in all twenty test problems because D is relatively smaller than the average length of time required to unload a batch of products from a receiving truck.

For each test problem in *Case 3* model, three solutions were obtained from the heuristic algorithm based on three strategies in *Phase I*. Of the three strategies, *strategy 3* (maximum fitness) performed the best, although it did find the worst solutions among the three strategies in a few problem instances. As a result, under real application it is advisable to run all three strategies to determine the best solution for a given problem instance. Even under normal algorithmic test situations, it is also suggested that all three strategies be run to ensure that the best solution is found in each case.

For the first set of twenty test problems with 75 units of delay time for a truck change operation, the average percentage deviation of the best compound solutions for the *Case 3* problem from the optimal solutions of the *Case 1* model is –1.39%. This implies the solutions obtained by the heuristic under the *Case 3* model are better, on the average, than the optimal solutions obtained from the *Case 1* model for the same problem scenario. For the second set of twenty test problems with 15 units of delay time for truck change operation, the average percentage deviation between the optimal solutions of the *Case 2* model and the compound heuristic solution of the *Case 3* problem is –3.19%. Again, the heuristic algorithm for the *Case 3* problem found superior solutions than the optimal solutions for the *Case 2* model on the average for the same problem scenarios. Overall, the compound heuristic algorithms for the *Case 3* model produced solutions that were superior to the optimal solutions for the *Cases 1* and *2* models on the average regardless of the truck change time. This implies the implementation of a flexible cross docking policy that allows trucks to make repeat visits to docks and allowing items to be routed to temporary storage is superior to policies that remove these flexibilities either partially or fully.

CHAPTER 7. CONCLUSIONS AND FUTURE RESEARCH

7.1 Conclusions

A cross docking operation involves multiple trucks (known as receiving trucks) that deliver products or items from suppliers to a warehouse and multiple trucks (known as shipping trucks) that ship items from the warehouse to customers. Based on customer demands, a receiving truck may have its items transferred to multiple shipping trucks. Similarly, a shipping truck can receive its consignments from multiple receiving trucks. A unique characteristic of a cross docking system is the absence or prohibition of long term storage of items at the warehouse. Items delivered to the warehouse from suppliers are shipped to customers as soon as possible without being placed in storage in the warehouse. The system can be operated with or without temporary storage. Ultimately, at the end of schedule period (e.g, one day), no item is left in the temporary storage.

Depending on the facility and operating conditions or strategies employed, it is possible to generate various cross docking models. In this research, thirty-two different models are suggested based on the number of docks at the site, the dock holding pattern for trucks, and the existence of temporary storage. Among the thirty-two models, three specific models of the cross docking systems are considered in this research. The three models are as follows:

Case 1 Model. There is temporary storage in front of the shipping dock. In this model, once a receiving truck or shipping truck pulls into a dock, it must stay at the dock until all of its items are unloaded if it is a receiving truck or all of its items are loaded if it is a shipping truck. A truck can come and leave the dock only once during a schedule period. Separate receiving and shipping docks are assumed.

Case 2 Model. In this model, there is no temporary storage in the warehouse or distribution center. However, both the receiving truck and the shipping truck can move in and out of the docks as many times as needed until their unloading or loading tasks are completed.

Case 3 Model. In this model, there is temporary storage in front of the shipping dock and both the receiving truck and the shipping truck can move in and out of the dock until their tasks are completed.

One of the objectives for cross docking systems is how well the trucks can be scheduled at the dock and how the items in receiving trucks can be allocated to the shipping trucks to optimize on some measure of system performance. The objective of this research is to find the best sequence for truck spotting for both receiving and shipping trucks to minimize total operation time or to maximize the throughput of the cross docking system. The product routing is also decided simultaneously as well as the spotting sequences of the receiving and shipping trucks.

7.1.1 *Case 1*

To solve the cross docking problem for the *Case 1* problem, five different approaches were developed; the mixed integer programming model, the complete enumeration of all possible sequences, the heuristic algorithm, the tabu search and the branch and bound method. The first two approaches were able to find the global optimal solution, but were not effective for solving medium to large size problems because of their computational intensity. Therefore, to increase solution efficiency, the heuristic algorithms were developed.

The main idea of the heuristic algorithm was to minimize the total number of products that pass through temporary storage. Because three strategies for selecting the associate receiving trucks and three strategies for selecting the shipping truck were developed, a total of nine heuristic algorithms were developed and tested for the *Case 1* problem. In order to test the performance of the heuristic algorithm, the twenty test problem sets were randomly generated. The problems were solved using the mixed integer programming model, the complete enumeration method, and the heuristic. Comparison of the solutions obtained from the heuristics with the optimal solutions indicated that the heuristic solutions were close to the optimal solutions. The heuristic solutions differed from the optimal solutions by an average of 1.80%; a value that clearly attests to the effectiveness of the heuristic.

As the fourth approach, the tabu search was implemented to further test the performance of the heuristic algorithm. In comparing the heuristic solutions with the tabu solutions, the tabu search outperformed the heuristic algorithm if the best out of ten different tabu solutions generated by using ten different random starting solutions for each problem set is selected. However, if only one random starting solution is used for the tabu search for each set of test problem and the final solutions obtained from the one single starting random solution were used for comparison with the heuristic solutions, then the heuristic algorithm outperformed the tabu search.

The last approach suggested for the *Case 1* problem was the branch and bound method. In implementing the branch and bound approach, the best heuristic solution obtained for each problem was used as the initial upper bound. The use of a good starting upper bound helped to reduce the computational time required to solve and obtain the optimal solutions to the problems. The computational time required by the branch and bound method was smaller than that required by the complete enumeration method.

7.1.2 *Case 2*

To solve the cross docking problem for the *Case 2* model, three approaches were developed; a mixed integer programming model, an integer programming model and a heuristic algorithm. The objective of the first mixed integer programming (Model I) model is to minimize makespan. For the Model I model, the number of variables and constraints grow exponentially as the number of receiving trucks, number of shipping trucks, and number of product types increase. Computationally, the approach is not effective for solving the test problems, including the smallest one. A different view point to the *Case 2* problem led to the development of a second integer program model (Model II). Model II is relatively simple. The objective of the integer programming model or Model II is to minimize the number of matching pairs of the receiving and shipping trucks. A receiving truck and a shipping truck are said to form a matching pair if there is material exchange between them. With the objective of minimizing the number of matching pairs, the number of variables and constraints are dramatically decreased. However, in spite of the reduction in the number variables and constraints, Model II was still found not to be effective for solving medium to large size problems because of computational time requirement. Therefore, to increase solution efficiency, heuristic algorithms were developed.

The third approach used heuristic algorithms. Six heuristic algorithms were developed and tested for the *Case 2* problem. The six heuristics employed different rules to determine the next shipping and receiving trucks to select or schedule. The six heuristics are based on the followings basic principles; 1) maximum flow between pairs, 2) maximum ratio between pairs, 3) maximum fitness between pairs, 4) maximum flow with priority assignment, 5) maximum ratio with priority assignment, and 6) maximum fitness with priority assignment. Of the six heuristic algorithms, heuristic algorithm 3 (maximum fitness) performed the best. It found the best solutions among the six heuristic solutions in fifteen out of twenty test problems. Overall, the heuristic algorithms produced solutions that were close to the global optimal solutions. For example, the range of percentage deviations from optimal for the twenty problem sets based on makespan minimization is 0%-9.90%. The overall average deviation from optimal is 3.49%.

For the *Case 2* problem, once the minimum number of matching pairs is found, then makespan is the same regardless of the order of selection of the pairs. The second integer programming model and the heuristic algorithms found the minimum number of matching pairs for the receiving and shipping trucks instead of directly finding the receiving and shipping truck sequences. However, the real interest is not the minimization of the number of matching pairs but the best spotting sequence for both receiving and shipping trucks. Therefore, there was the need to develop a method that would convert the matching pairs to near optimal sequences for the receiving and shipping trucks.

In order to find the sequences for both receiving and shipping trucks, where the number of matching pairs is given from the previous solution, solution approaches to minimize the mean flow time for receiving and shipping trucks in the distribution center were developed. The flow time of a receiving (shipping) truck is defined as the time interval between the time the truck first unloads (loads) an item to the time it unloads (loads) its last item or unit.

To solve the problem, two approaches were used; the complete enumeration method and the tabu search method. Using the complete enumeration method, the optimal solutions were found in eleven out of the twenty test problems. Solutions to all the twenty problems could not be obtained because of the high computational time required. On the other hand, the tabu search found the solutions very quickly. It found the solutions within twenty seconds for all twenty problems. The performance of the tabu search was very successful. The tabu search found the optimal solutions in all the eleven problems whose optimal solutions are known. To reduce the computational time of the tabu search, the net change equation of the flow time was developed and used instead of directly calculating the mean flow time for each adjacent neighborhood interchange.

7.1.3 *Case 3*

To solve the cross docking problem for the *Case 3* problem, two approaches were developed; the mixed integer programming model and the heuristic algorithm. Even though the *Case 3* problem could be modeled mathematically, it is difficult to solve because of the large number of variables and constraints present as the number of receiving trucks, number of shipping trucks, and the number of product types increase. Therefore, the second approach was developed to solve the problems. The second approach used some heuristic algorithms.

The heuristic algorithm developed for the *Case 3* problem consists of two phases. In *Phase I*, product routing is determined. The initial receiving and shipping truck sequences are also created in *Phase I*. In the schedule of *Phase I*, no products are sent to temporary storage. In *Phase II*, the algorithm searches for improved solution by allowing items to be sent to temporary storage instead of changing the current receiving truck at the dock if that would reduce the makespan. If a certain condition that decreases the makespan is met, an appropriate quantity of the products is sent to temporary storage instead of changing the receiving trucks at the dock. The current spotting sequence is appropriately modified to reflect the number of truck changes at the dock. The *Phase II* search is continued until the schedule does not identify any conditions that would decrease the makespan. Based on the strategy for finding the best associate receiving trucks for each unscheduled shipping truck in *Phase I*, three heuristic algorithms were developed; 1) maximum flow, 2) maximum ratio, and 3) maximum fitness. Among the three heuristic algorithms, the average performance of strategy 3 (maximum fitness) was the best regardless of truck change time. However, strategy 3 also found the worst solution among the three strategies in a few problem cases. Therefore, for actual implementation, it is suggested that all three strategies of the heuristic be applied and to choose the best solution found by the three strategies as the adopted solution. Overall, the heuristic algorithm for *Case 3* produced solutions that were better than the optimal solutions for *Cases 1* and *2* on the average regardless of truck change time.

7.2 Future Research

Because research in cross docking is at its infancy, tremendous opportunities exist in the future to exploit the vastly untapped possible extensions to this research. Here are the representative future research areas that can be extended from this research.

1. All three models studied in this research assumed that a warehouse or a distribution center has only one receiving dock and one shipping dock, respectively. The development of models with multiple receiving docks and multiple shipping docks will constitute an excellent future research area.
2. So far, temporary storage is assumed to have infinite capacity. However, the capacity of the temporary storage is often constrained in practice. In this case, a different operating strategy that recognizes the capacity limitation may be needed. Development of an operating strategy that recognizes the presence of a capacity constrained temporary storage is another possible extension to the current research.
3. In some distribution centers, both cross docking operation and a regular warehouse for long term material storage may co-exist together. In those distribution centers, some products unloaded from receiving trucks may transfer to the warehouse section for long-term storage while others are directly transferred from the receiving trucks to the shipping docks as in cross docking systems. Similarly, a shipping truck may load some of its needed products directly from receiving trucks and some of others from the long-term storage warehouse. Development of a model for this kind of operating scenario is another possible future extension of the current research.
4. While this research is focused on activities within one warehouse, the greatest opportunities in both research and economy of scale appears to be in the integration of multiple warehouses. One of the future research areas is the development of models and solution algorithms that considers not only the operations within a warehouse but also the integration of the whole operations in a network consisting of multiple supply chain members that are distributed over a wide geographic area. Such network problem can be

addressed based on physical material flow management or information management. In order to facilitate a real-time access to the network-wide solutions developed, the deployment of the models and the solution procedures on Internet can be considered in future research.

APPENDIX A. STEP BY STEP PROCEDURE FOR SOLVING HEURISTIC ALGORITHM FOR THE *CASE 1* PROBLEM

The following problem is used to demonstrate the use of the heuristic algorithm for the *Case 1* problem. The problem has a set of four receiving trucks and three shipping trucks. The number of product types is four. All information about receiving trucks and shipping trucks is presented in Table A-1. It is assumed that all types of loading and unloading times for all types of products are the same and they are one unit of time in duration. Truck change time takes 75 units of time and transferring time of products from the receiving dock to the shipping dock takes 100 units of time.

Assume that the first strategy for the associate receiving truck selection strategy is chosen for this demonstration. For the shipping truck selection strategy, the first strategy is also selected.

1. *Associate Receiving Truck Selection Strategy 1:*

- Choose the receiving truck that transfers the smallest number of products from a receiving truck to temporary storage.

Table A-1. Example Set to Illustrate the Heuristic Algorithm for the *Case 1* Problem

Receiving Truck

Truck	Product	Quantity
1	1	100
	3	50
2	1	100
	2	50
	3	100
3	1	100
	2	40
	4	60
4	2	100
	4	200

Shipping Truck

Truck	Product	Quantity
1	1	100
	2	100
2	1	120
	3	110
	4	160
3	1	80
	2	90
	3	40
	4	100

2. *Shipping Truck Selection Strategy 1:*

Choose the shipping truck and its associated receiving trucks that transfer the smallest number of products from the associate receiving trucks to temporary storage.

The following solution procedure shows the entire steps for solving the problem with the heuristic algorithm for the *Case 1* problem.

<u>*STEP 1*</u>

Set $T^r = \varnothing$, $T^s = \varnothing$, $U^r = \{t^r_1, t^r_2, t^r_3, t^r_4\}$ and $U^s = \{t^s_1, t^s_2, t^s_3\}$. Set $t_1 = t_2 = t_3 = t_4 = 0$.

<u>*STEP 2*</u>

For shipping truck t^s_1,

<u>*2a*</u> $A^R_1 = \varnothing$, $p^{AT}_1 = 0$, $p^{AS}_1 = 200$ (=100+100+0+0).

<u>*2b*</u> $T^r = \varnothing$. $s'_{1k} \leftarrow s_{1k}$, for k = 1, 2, 3 and 4. Go to *2d* in *Step 2*.

<u>*2d*</u> $\sum_{k=1}^{4}\left[\max\{s'_{1k} - t_k, 0\}\right] = 200$. Go to *2e* in *Step 2*.

<u>*2e*</u>

For t^r_1 : $p^{RT}_{11} = 50$ (50 units of product type 3)

$p^{RS}_{11} = 100$ (100 units of product type 1)

For t^r_2 : $p^{RT}_{21} = 100$ (100 units of product type 3)

$p^{RS}_{21} = 150$ (100 units of product type 1 and 50 units of product type 2)

For t^r_3 : $p^{RT}_{31} = 60$ (60 units of product type 4)

$p^{RS}_{31} = 140$ (100 units of product type 1 and 40 units of product type 2)

For t^r_4 : $p^{RT}_{41} = 200$ (200 units of product type 4)

$p^{RS}_{41} = 100$ (100 units of product type 2)

To calculate p^{RT}_{i1}, *equation (4-15)* was used. For p^{RS}_{i1}, *equation (4-16)* was used.

<u>*2f*</u> Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_1. $A^R_1 = \{t^r_1\}$.

Note that if the associate receiving truck selection strategy 2 is applied, receiving truck t^r_2 is selected here instead of receiving truck t^r_1. If the associate receiving truck selection strategy 3 is applied, receiving truck t^r_3 is selected here instead of receiving truck t^r_1. As it can be seen here, the associated receiving trucks can be formed differently for the same shipping truck based on the associate receiving truck selection strategy.

2g $s'_{11} = s'_{13} = s'_{14} = 0$ and $s'_{12} = 100$. Since $\sum_{k=1}^{4} s'_{1k} = 100$, go to *2d* in *Step 2.*

2d $\sum_{k=1}^{4}[\max\{s'_{1k} - t_k, 0\}] = 100$. Go to *2e* in *Step 2.*

2e

For t^r_2 : $p^{RT}_{21} = 200$ (100 units of product type 1 and 100 units of product type 3)

$p^{RS}_{21} = 50$ (50 units of product type 2)

For t^r_3 : $p^{RT}_{31} = 160$ (100 units of product type 1 and 60 units of product type 4)

$p^{RS}_{31} = 40$ (40 units of product type 2)

For t^r_4 : $p^{RT}_{41} = 40$ (40 units of product type 4)

$p^{RS}_{41} = 100$ (100 units of product type 2)

To calculate p^{RT}_{21} and p^{RT}_{31} *equation (4-15)* was used. However, *equation (4-14)* was used to calculate p^{RT}_{41}. Note that shipping truck t^s_1 is filled all of its needed products with receiving trucks t^r_1 and t^r_4. Therefore, it looks like no products are sent to temporary storage. (i.e. it looks like $p^{RT}_{41} = 0$). However, if receiving truck t^r_4 is scheduled in the second position of the receiving truck sequence, a total of 40 units of product type 4 needs to be sent to temporary storage when next scheduled shipping truck is t^s_2. If the next scheduled shipping truck is t^s_3, a total of 100 units of product type 4 needs to be sent to temporary storage. Therefore $p^{RT}_{41} = 40$ presents the least number of products that transfer to temporary storage when receiving truck t^r_4 is scheduled in the second position in the receiving truck sequence. It can be calculated from *equation (4-14)*. To calculate p^{RS}_{i1}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_4. $A^R_1 = \{t^r_1, t^r_4\}$.

2g $s'_{11} = s'_{12} = s'_{13} = s'_{14} = 0$. Since $\sum_{k=1}^{N} s'_{1k} = 0$, go to *2h* in *Step 2.*

2h Receiving trucks t^s_2 and t^s_3 do not have their associated receiving trucks. Go to the beginning of the *Step 2.*

For shipping truck 2, t^s_2,

2a $A^R_2 = \varnothing$, $p^{AT}_2 = 0$, $p^{AS}_2 = 390$ (=120+0+110+160).

2b $T^r = \varnothing$. $s'_{2k} \leftarrow s_{2k}$, for k = 1, 2, 3 and 4. Go to *2d* in *Step 2*.

2d $\sum_{k=1}^{4}[\max\{s'_{2k} - t_k, 0\}] = 390$. Go to *2e* in *Step 2*.

2e

For t^r_1 : $p^{RT}_{12} = 0$

$p^{RS}_{12} = 150$ (100 units of product type 1 and 50 units of product type 3)

For t^r_2 : $p^{RT}_{22} = 50$ (50 units of product type 2)

$p^{RS}_{22} = 200$ (100 units of product type 1 and 100 units of product type 3)

For t^r_3 : $p^{RT}_{32} = 40$ (40 units of product type 2)

$p^{RS}_{32} = 160$ (100 units of product type 1 and 60 units of product type 4)

For t^r_4 : $p^{RT}_{42} = 140$ (100 units of product type 2 and 40 units of product type 4)

$p^{RS}_{42} = 160$ (160 units of product type 4)

To calculate p^{RT}_{i2}, *equation (4-15)* was used. For p^{RS}_{i2}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_1. $A^R_2 = \{t^r_1\}$.

2g $s'_{21} = 20$, $s'_{22} = 0$, $s'_{23} = 60$ and $s'_{24} = 160$. Since $\sum_{k=1}^{4} s'_{2k} = 240$, go to *2d* in *Step 2*.

2d $\sum_{k=1}^{4}[\max\{s'_{2k} - t_k, 0\}] = 240$. Go to *2e* in *Step 2*.

2e

For t^r_2 : $p^{RT}_{22} = 170$ (80 units of product type 1, 50 units of product type 2 and 40 units of product type 3)

$p^{RS}_{22} = 80$ (20 units of product type 1 and 60 units of product type 3)

For t^r_3 : $p^{RT}_{32} = 120$ (80 units of product type 1 and 40 units of product type 2)

$p^{RS}_{32} = 80$ (20 units of product type 1 and 60 units of product type 4)

For t^r_4 : $p^{RT}_{42} = 140$ (100 units of product type 2 and 40 units of product type 4)

$p^{RS}_{42} = 160$ (160 units of product type 4)

To calculate p^{RT}_{i2}, *equation (4-15)* was used. For p^{RS}_{i2}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t'_3. $A^R_2 = \{t'_1, t'_3\}$.

2g $s'_{21} = s'_{22} = 0$, $s'_{23} = 60$ and $s'_{24} = 100$. Since $\sum_{k=1}^{4} s'_{2k} = 160$, go to *2d* in *Step 2*.

2d $\sum_{k=1}^{4} [\max\{s'_{2k} - t_k, 0\}] = 160$. Go to *2e* in *Step 2*.

2e

For t'_2 : $p^{RT}_{22} = 190$ (100 units of product type 1, 50 units of product type 2 and 40 units of product type 3)

$p^{RS}_{22} = 60$ (60 units of product type 3)

For t'_4 : $p^{RT}_{42} = 200$ (100 units of product type 2 and 100 units of product type 4)

$p^{RS}_{42} = 100$ (100 units of product type 4)

To calculate p^{RT}_{i2}, *equation (4-15)* was used. For p^{RS}_{i2}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t'_2. $A^R_2 = \{t'_1, t'_3, t'_2\}$.

2g $s'_{21} = s'_{22} = s'_{23} = 0$ and $s'_{24} = 100$. Since $\sum_{k=1}^{4} s'_{2k} = 100$, go to *2d* in *Step 2*.

2d $\sum_{k=1}^{4} [\max\{s'_{2k} - t_k, 0\}] = 100$. Go to *2e* in *Step 2*.

2e

For t'_4 : $p^{RT}_{42} = 0$

$p^{RS}_{42} = 100$ (100 units of product type 4)

equation (4-14) was used to calculate p^{RT}_{42}. To calculate p^{RS}_{i1}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t'_4. $A^R_2 = \{t'_1, t'_3, t'_2, t'_4\}$.

2g $s'_{21} = s'_{22} = s'_{23} = s'_{24} = 0$. Since $\sum_{k=1}^{4} s'_{2k} = 0$, go to *2h* in *Step 2*.

2h t^s_3 does not have its associated receiving trucks. Go to the beginning of the *Step 2*.

For shipping truck 3, t^s_3,

<u>2a</u> $A^s_3 = \varnothing$, $p^{AT}_3 = 0$, $p^{AS}_3 = 310$ (=80+90+40+100).

<u>2b</u> $T^r = \{\ \}$. $s'_{3k} \leftarrow s_{3k}$, for k = 1, 2, 3 and 4. Go to *2d* in *Step 2*.

<u>2d</u> $\sum_{k=1}^{4}[\max\{s'_{3k} - t_k, 0\}] = 310$. Go to *2e* in *Step 2*.

<u>2e</u>

For t^r_1 : $p^{RT}_{13} = 30$ (20 units of product type 1 and 10 units of product type 3)

$p^{RS}_{13} = 120$ (80 units of product type 1 and 40 units of product type 3)

For t^r_2 : $p^{RT}_{23} = 80$ (20 units of product type 1 and 60 units of product type 3)

$p^{RS}_{23} = 170$ (80 units of product type 1, 50 units of product type 2 and 40 units of product type 3)

For t^r_3 : $p^{RT}_{33} = 20$ (20 units of product type 1)

$p^{RS}_{33} = 180$ (80 units of product type 1, 40 units of product type 2 and 60 units of product type 4)

For t^r_4 : $p^{RT}_{43} = 110$ (10 units of product type 2 and 100 units of product type 4)

$p^{RS}_{43} = 190$ (90 units of product type 2 and 100 units of product type 4)

To calculate p^{RT}_{i3}, *equation (4-15)* was used. For p^{RS}_{i3}, *equation (4-16)* was used.

<u>2f</u> Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_3. $A^R_3 = \{t^r_3\}$.

<u>2g</u> $s'_{31} = 0$, $s'_{32} = 50$, $s'_{33} = 40$ and $s'_{34} = 40$. Since $\sum_{k=1}^{4} s'_{3k} = 130$, go to *2d* in *Step 2*.

<u>2d</u> $\sum_{k=1}^{4}[\max\{s'_{3k} - t_k, 0\}] = 130$. Go to *2e* in *Step 2*.

<u>2e</u>

For t^r_1 : $p^{RT}_{13} = 110$ (100 units of product type 1 and 10 units of product type 3)

$p^{RS}_{13} = 40$ (40 units of product type 3)

For t^r_2 : $p^{RT}_{23} = 160$ (100 units of product type 1 and 60 units of product type 3)

$p^{RS}_{23} = 90$ (50 units of product type 2 and 40 units of product type 3)

For t^r_4 : $p^{RT}_{43} = 210$ (50 units of product type 2 and 160 units of product type 4)

$p^{RS}_{43} = 90$ (50 units of product type 2 and 40 units of product type 4)

To calculate p^{RT}_{i3}, *equation (4-15)* was used. For p^{RS}_{i3}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_1. $A^R_3 = \{t^r_3, t^r_1\}$.

2g $s'_{31} = 0$, $s'_{32} = 50$, $s'_{33} = 0$ and $s'_{34} = 40$. Since $\sum_{k=1}^{4} s'_{3k} = 90$, go to *2d* in *Step 2*.

2d $\sum_{k=1}^{4}[\max\{s'_{3k} - t_k, 0\}] = 90$. Go to *2e* in *Step 2*.

2e

For t^r_2 : $p^{RT}_{23} = 200$ (100 units of product type 1 and 100 units of product type 3)

$p^{RS}_{23} = 50$ (50 units of product type 2)

For t^r_4 : $p^{RT}_{43} = 50$ (50 units of product type 2)

$p^{RS}_{43} = 90$ (50 units of product type 2 and 40 units of product type 4)

To calculate p^{RT}_{23} *equation (4-15)* was used. However, *equation (4-14)* was used to calculate p^{RT}_{43}. If the next scheduled shipping truck is t^s_2, a total of 50 units of product type 2 needs to be sent to temporary storage. For p^{RS}_{i3}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_4. $A^R_3 = \{t^r_3, t^r_1, t^r_4\}$.

2g $s'_{31} = s'_{32} = s'_{33} = s'_{34} = 0$. Since $\sum_{k=1}^{4} s'_{3k} = 0$, go to *2h* in *Step 2*.

2h All shipping trucks have their associated receiving trucks. Go to *Step 3*.

STEP 3

For t^s_1 and $A^R_1 = \{t^r_1, t^r_4\}$,

$p^{AT}_1 = 90$ (50 units of product type 3 and 40 units of product type 4)

$p^{AS}_1 = 200$ (100 units of product type 1 and 100 units of product type 2)

$p^{TM}_1 = 365$ $(= p^{AS}_1 + p^{AT}_1 + (z-1)D)$ (In this case, $z = 2$ and $D = 75$)

For t^s_2 and $A^R_2 = \{t^r_1, t^r_3, t^r_2, t^r_4\}$,

$p^{AT}_2 = 320$ (180 units of product type 1, 100 units of product type 2 and 40 units of product type 3)

$p^{AS}{}_{2}$ = 390 (120 units of product type 1, 110 units of product type 3 and 160 units of product type 4)

$p^{TM}{}_{2} = 935$ (= $p^{AS}{}_{2} + p^{AT}{}_{2} + (z-1)D$) (In this case, $z = 4$ and $D = 75$)

For $t^{s}{}_{3}$ and $A^{R}{}_{3} = \{t^{r}{}_{3}, t^{r}{}_{1}, t^{r}{}_{4}\}$,

$p^{AT}{}_{3}$ = 180 (120 units of product type 1, 50 units of product type 2 and 10 units of product type 3)

$p^{AS}{}_{3}$ = 310 (80 units of product type 1, 90 units of product type 2, 40 units of product type 3 and 100 units of product type 4)

$p^{TM}{}_{3} = 640$ (= $p^{AS}{}_{3} + p^{AT}{}_{3} + (z-1)D$) (In this case, $z = 3$ and $D = 75$)

STEP 4

Because the shipping truck selection strategy 1 is used here, choose shipping truck $t^{s}{}_{1}$,

$U^{s} = \{t^{s}{}_{2}, t^{s}{}_{3}\}$. $T^{s} = \{t^{s}{}_{1}\}$.

STEP 5

$U^{r} = \{t^{r}{}_{2}, t^{r}{}_{3}\}$. $T^{r} = \{t^{r}{}_{1}, t^{r}{}_{4}\}$.

STEP 6

$t^{r}{}_{L} = t^{r}{}_{4}$. $r_{l1} = r_{l2} = r_{l3} = 0$ and $r_{l4} = 200$. $t_1 = t_2 = 0$, $t_3 = 50$ and $t_4 = 40$.

STEP 7

$U^{r} = \{t^{r}{}_{2}, t^{r}{}_{3}\}$. $U^{s} = \{t^{s}{}_{2}, t^{s}{}_{3}\}$. Go to *Step 2*.

STEP 2

For shipping truck 2, $t^{s}{}_{2}$,

2a $A^{R}{}_{2} = \varnothing$, $p^{AT}{}_{2} = 0$, $p^{AS}{}_{2} = 390$ (=120+0+110+160).

2b $s'_{21} = 120$, $s'_{22} = 0$, $s'_{23} = 110$ and $s'_{24} = 0$. $A^{R}{}_{2} = \{ t^{r}{}_{4} \}$.

2c Because $\sum_{k=1}^{4} s'_{2k} = 230$, calculate $p^{AT}{}_{2}$.

$p^{AT}{}_{2}$ = 40 (40 units of product type 4). Go to *2d* in *Step 2*.

2d Because $\sum_{k=1}^{4}[\max\{s'_{2k} - t_k, 0\}] = 180$ (120 units of product type 1 and 60 units of product type 3), go to *2e* in *Step 2*.

2e

For t^r_2 : $p^{RT}_{22} = 50$ (50 units of product type 2)

$p^{RS}_{22} = 200$ (100 units of product type 1 and 100 units of product type 3)

For t^r_3 : $p^{RT}_{32} = 100$ (40 units of product type 2 and 60 units of product type 4)

$p^{RS}_{32} = 100$ (100 units of product type 1)

To calculate p^{RT}_{i2}, *equation (4-15)* was used. For p^{RS}_{i2}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_2. $A^R_2 = \{t^r_2\}$.

2g $s'_{21} = 20$, $s'_{22} = 0$, $s'_{23} = 10$ and $s'_{24} = 0$. Since $\sum_{k=1}^{4} s'_{2k} = 30$, go to *2d* in *Step 2*.

2d Because $\sum_{k=1}^{4} [\max\{s'_{2k} - t_k, 0\}] = 20$ (20 units of product type 1), go to *2e* in *Step 2*.

2e

For t^r_3 : $p^{RT}_{32} = 0$

$p^{RS}_{32} = 20$ (20 units of product type 1)

To calculate p^{RT}_{i2}, *equation (4-14)* was used. For p^{RS}_{i2}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_3. $A^R_2 = \{t^r_2, t^r_3\}$.

2g $s'_{21} = 0$, $s'_{22} = 0$, $s'_{23} = 10$ and $s'_{24} = 0$. Since $\sum_{k=1}^{4} s'_{2k} = 10$, go to *2d* in *Step 2*.

2d Because $\sum_{k=1}^{4} [\max\{s'_{2k} - t_k, 0\}] = 0$, go to *2h* in *Step 2*.

2h Shipping truck t^s_3 does not have its associated receiving trucks. Go to the beginning of the *Step 2*.

For shipping truck 3, t^s_3,

2a $A^R_3 = \varnothing$, $p^{AT}_3 = 0$, $p^{AS}_3 = 310$ (=80+90+40+100).

2b $s'_{11} = 80$, $s'_{22} = 90$, $s'_{23} = 40$ and $s'_{24} = 0$. $A^R_3 = \{ t^r_4 \}$.

2c Because $\sum_{k=1}^{4} s'_{3k} = 210$, calculate p^{AT}_3.

$p^{AT}_3 = 100$ (100 units of product type 4). Go to *2d* in *Step 2*.

2d Because $\sum_{k=1}^{4}[\max\{s'_{3k} - t_k, 0\}] = 170$ (80 units of product type 1 and 90 units of product type 2), go to *2e* in *Step 2*.

2e

For t^r_2 : p^{RT}_{23} = 80 (20 units of product type 1 and 60 units of product type 3)

p^{RS}_{23} = 170 (80 units of product type 1, 50 units of product type 2 and 40 units of product type 3)

For t^r_3 : p^{RT}_{33} = 80 (20 units of product type 1 and 60 units of product type 4)

p^{RS}_{33} = 120 (80 units of product type 1 and 40 units of product type 2)

To calculate p^{RT}_{i2}, *equation (4-15)* was used. For p^{RS}_{i2}, *equation (4-16)* was used.

2f Because the associate receiving truck selection strategy 1 is used, choose the receiving truck t^r_2. $A^R_3 = \{t^r_2\}$.

2g $s'_{31} = s'_{33} = s'_{34} = 0$ and $s'_{32} = 50$. Since $\sum_{k=1}^{4} s'_{3k} = 50$, go to *2d* in *Step 2*.

2d Because $\sum_{k=1}^{4}[\max\{s'_{3k} - t_k, 0\}] = 0$, go to *2h* in *Step 2*.

2h All shipping trucks have their associated receiving trucks. Go to *Step 3*.

STEP 3

For t^s_2 and $A^R_2 = \{t^r_2, t^r_3\}$,

p^{AT}_2 = 90 (50 units of product type 2 and 40 units of product type 3)

p^{AS}_2 = 390 (120 units of product type 1, 110 units of product type 3 and 160 units of product type 4)

p^{TM}_2 = 630 (= $p^{AS}_2 + p^{AT}_2 + (z-1)D$) (In this case, z = 3 including the last scheduled receiving truck and D = 75)

For t^s_3 and $A^R_3 = \{t^r_2\}$,

p^{AT}_3 = 180 (20 units of product type 1, 60 units of product type 3 and 100 units of product type 4)

p^{AS}_3 = 310 (80 units of product type 1, 90 units of product type 2, 40 units of product type 3, and 100 units of product type 4)

$p^{TM}{}_3$ = 565 (= $p^{AS}{}_3 + p^{AT}{}_3 + (z\text{-}1)D$) (In this case, z = 2 including the last scheduled receiving truck and D = 75)

<u>*STEP 4*</u>

Because the shipping truck selection strategy 1 is used here, choose shipping truck $t^s{}_2$,

$U^s = \{t^s{}_3\}$. $T^s = \{t^s{}_1, t^s{}_2\}$.

<u>*STEP 5*</u>

$U^r = \{\ \}$. $T^r = \{t^r{}_1, t^r{}_4, t^r{}_2, t^r{}_3\}$.

<u>*STEP 6*</u>

$t^r{}_L = t^r{}_3$. $r_{l1} = 80$, $r_{l2} = 40$, $r_{l3} = 0$ and $r_{l4} = 60$. $t_1 = 0$, $t_2 = 50$, $t_3 = 40$ and $t_4 = 40$.

<u>*STEP 7*</u>

$U^r = \{\ \}$. $U^s = \{t^s{}_3\}$. Note that the only unscheduled shipping truck is $t^s{}_3$. Therefore, the shipping truck $t^s{}_3$ can be placed to the end of set T^s and the heuristic algorithm can be finished here. However, the algorithm will be followed further in order to explain the remaining steps of the algorithm until it finishes. Then the next step will be *Step 2*.

<u>*STEP 2*</u>

For shipping truck 3, $t^s{}_3$,

<u>*2a*</u> $A^s{}_3 = \varnothing$, $p^{AT}{}_3 = 0$, $p^{AS}{}_3 = 310$ (=80+90+40+100).

<u>*2b*</u> $s'_{11} = 0$, $s'_{22} = 50$, $s'_{23} = 40$ and $s'_{24} = 40$. $A^R{}_3 = \{\ t^r{}_3\ \}$.

<u>*2c*</u> Because $\sum_{k=1}^{4} s'_{3k} = 130$, calculate $p^{AT}{}_3$.

$p^{AT}{}_3 = 0$. Go to *2d* in *Step 2*.

<u>*2d*</u> Because $\sum_{k=1}^{4}\left[\max\{s'_{3k} - t_k, 0\}\right] = 0$, go to *2h* in *Step 2*.

<u>*2h*</u> All shipping trucks have their associated receiving trucks. Go to *Step 3*.

<u>*STEP 3*</u>

For $t^s{}_3$ and $A^R{}_3 = \{\ \}$,

$p^{AT}{}_3 = 0$

$p^{AS}{}_3 = 310$ (80 units of product type 1, 90 units of product type 2, 40 units of product type 3, and 100 units of product type 4)

$p^{TM}{}_3 = 310$ ($= p^{AS}{}_3 + p^{AT}{}_3$)

STEP 4

Because the shipping truck selection strategy 1 is used here, choose shipping truck t^s_3,

$U^s = \{\ \}$. $T^s = \{t^s_1, t^s_2, t^s_3\}$.

STEP 5

$U^r = \{\ \}$. $T^r = \{t^r_1, t^r_4, t^r_2, t^r_3\}$.

STEP 6

$t^r_l = t^r_3$. $r_{l1} = r_{l2} = r_{l3} = r_{l4} = 0$. $t_1 = t_2 = t_3 = t_4 = 0$.

STEP 7

If $U^r = U^s = \varnothing$, stop. All sequences for receiving trucks and shipping trucks are found. T^r presents the sequence for receiving trucks. T^s shows the sequence for shipping trucks.

$$T^r = \{t^r_1, t^r_4, t^r_2, t^r_3\}.$$

$$T^s = \{t^s_1, t^s_2, t^s_3\}.$$

Total number of products that pass through temporary storage is 180 units (50 units of product type 2, 90 units of product type 3 and 40 units of product type 4).

END OF HEURISTIC ALGORITHM

APPENDIX B. TWENTY TEST PROBLEM SETS

Table B-1. Test Set 1

Receiving Truck

Truck	Product	Quantity
1	1	48
	2	36
	3	84
	4	72
2	1	89
	2	127
	3	64
3	1	75
	2	105
	3	15
	4	15
4	2	260

Shipping Truck

Truck	Product	Quantity
1	1	151
	4	87
2	2	106
	3	33
3	2	264
4	1	61
	2	132
5	2	26
	3	130

Table B-2. Test Set 2

Receiving Truck

Truck	Product	Quantity
1	1	48
	2	85
	6	97
2	2	57
	4	47
	5	66
3	1	80
	3	70
	5	70
	6	10
4	1	18
	2	61
	4	43
	5	30
	6	18
5	1	76
	2	10
	3	43
	4	75
	5	26

Shipping Truck

Truck	Product	Quantity
1	1	74
	3	75
	4	99
	5	72
	6	23
2	1	123
	2	124
3	6	63
4	1	25
	2	89
	3	38
	4	66
	5	120
	6	39

Table B-3. Test Set 3

Receiving Truck

Truck	Product	Quantity
1	1	14
	2	28
	3	55
	4	41
	5	96
	6	48
	7	28
	8	28
2	1	116
	2	41
	3	26
	4	50
	8	39
3	2	38
	3	64
	4	64
	8	114

Shipping Truck

Truck	Product	Quantity
1	2	50
	5	32
	6	24
	7	13
	8	102
2	1	130
	3	145
	6	24
3	2	57
	4	155
	5	64
	7	15
	8	79

Table B-4. Test Set 4

Receiving Truck

Truck	Product	Quantity
1	1	39
	2	6
	3	6
	4	22
	5	57
	6	22
	7	39
	8	39
2	1	71
	4	71
	6	78
3	3	160
4	2	40
	3	25
	4	45
	5	50
5	1	86
	5	14
	7	72
	8	58

Shipping Truck

Truck	Product	Quantity
1	2	12
	8	42
2	2	4
	3	75
	4	46
	5	50
	6	60
	8	9
3	3	41
	4	31
	5	20
	7	111
	8	18
4	1	103
	2	13
	3	75
	4	61
	5	30
	8	28
5	1	93
	2	17
	5	21
	6	40

Table B-5. Test Set 5

Receiving Truck

Truck	Product	Quantity
1	1	170
2	1	6
	2	6
	3	19
	4	50
	5	38
	6	6
	7	19
	8	56
3	1	49
	2	31
	3	60
	6	12
	7	37
	8	31
4	5	143
	7	47
5	4	58
	5	36
	7	72
	8	14

Shipping Truck

Truck	Product	Quantity
1	1	75
	2	12
	3	59
	6	9
	7	98
	8	40
2	1	150
	5	217
3	2	25
	3	20
	4	108
	6	9
	7	77
	8	61

Table B-6. Test Set 6

Receiving Truck

Truck	Product	Quantity
1	1	22
	2	22
	3	66
	4	37
	5	73
2	2	100
	3	60
	4	90
3	4	217
	5	43
4	1	11
	2	89
	3	22
	4	101
	5	67

Shipping Truck

Truck	Product	Quantity
1	1	22
2	4	159
	5	133
3	1	3
	3	92
	4	159
	5	33
4	1	8
	2	211
	3	56
	4	127
	5	17

Table B-7. Test Set 7

Receiving Truck

Truck	Product	Quantity
1	1	14
	2	69
	3	28
	5	69
2	1	50
	4	40
	6	70
3	2	190
4	1	23
	3	115
	5	92
5	3	44
	5	66
	6	110

Shipping Truck

Truck	Product	Quantity
1	1	50
	3	36
	6	95
2	2	194
	3	53
	5	62
	6	74
3	1	37
	2	65
	3	18
	4	40
	5	165
	6	11
4	3	80

Table B-8. Test Set 8

Receiving Truck

Truck	Product	Quantity
1	1	64
	2	75
	3	75
	7	96
2	3	300
3	1	57
	2	72
	4	36
	5	65
	6	50

Shipping Truck

Truck	Product	Quantity
1	2	38
	3	113
	7	22
2	2	58
	5	50
	6	12
	7	12
3	1	61
	2	19
	3	131
	4	11
	5	10
	7	20
4	1	40
	4	18
	5	5
	6	26
	7	17
5	1	20
	2	32
	3	131
	4	7
	6	12
	7	25

Table B-9. Test Set 9

Receiving Truck

Truck	Product	Quantity
1	2	32
	3	18
	4	45
	5	32
	6	41
	8	32
2	1	58
	2	52
	5	47
	6	35
	7	12
	8	46
3	1	187
	5	53
4	3	111
	4	99

Shipping Truck

Truck	Product	Quantity
1	1	245
	2	12
	4	52
	5	37
	6	33
	7	6
2	2	42
	4	92
	5	26
	6	43
3	5	43
	8	78
4	2	30
	3	129
	5	26
	7	6

Table B-10. Test Set 10

Receiving Truck

Truck	Product	Quantity
1	1	64
	2	58
	3	19
	4	38
	5	19
	6	58
	7	19
	8	7
	9	58
2	5	132
	9	118
3	2	49
	7	97
	8	49
	9	145

Shipping Truck

Truck	Product	Quantity
1	4	38
	6	29
	8	28
2	5	151
	6	10
	7	12
	8	28
3	2	41
	6	19
	7	61
	9	229
4	1	64
	2	66
	3	19
	7	43
	9	92

Table B-11. Test Set 11

Receiving Truck

Truck	Product	Quantity
1	1	50
	2	25
	3	123
	4	25
	6	37
2	1	73
	2	64
	3	37
	4	46
	5	73
	6	37
3	2	360
4	2	29
	3	118
	4	59
	5	74
5	2	390

Shipping Truck

Truck	Product	Quantity
1	1	18
	2	200
	3	14
	4	91
	5	19
	6	33
2	1	35
	2	134
	3	113
	4	39
	5	20
	6	41
3	2	534
4	1	70
	3	151
	5	108

Table B-12. Test Set 12

Receiving Truck

Truck	Product	Quantity
1	1	83
	2	46
	3	83
	4	83
	6	18
	8	47
2	7	340
3	2	129
	3	113
	4	97
	8	31
4	1	47
	3	94
	6	109
5	3	78
	5	61
	6	78
	8	43
6	1	25
	2	50
	4	59
	5	76
	6	59
	7	76
	8	25

Shipping Truck

Truck	Product	Quantity
1	7	416
	8	46
2	2	112
	4	119
	5	86
	6	185
	8	54
3	2	113
	4	120
	5	51
	6	79
	8	46
4	1	155
	3	368

Table B-13. Test Set 13

Receiving Truck

Truck	Product	Quantity
1	1	293
	7	37
2	3	310
3	1	75
	2	94
	5	57
	6	19
	7	85
4	1	74
	2	74
	3	15
	5	52
	6	67
	7	45
	8	23
5	2	28
	3	19
	4	84
	5	84
	6	28
	7	47

Shipping Truck

Truck	Product	Quantity
1	7	100
2	1	368
	2	41
	4	42
	5	48
	6	24
	8	13
3	6	57
	7	97
4	1	74
	2	62
	5	145
	6	17
	7	8
5	3	344
6	2	93
	4	42
	6	16
	7	9
	8	10

Table B-14. Test Set 14

Receiving Truck

Truck	Product	Quantity
1	3	111
	5	157
	8	32
2	6	136
	8	204
3	1	23
	3	101
	4	68
	5	34
	6	44
4	1	32
	2	136
	4	95
	5	11
	6	32
	7	32
	8	42
5	1	29
	2	48
	3	29
	4	66
	5	93
	6	86
	7	10
	8	29

Shipping Truck

Truck	Product	Quantity
1	1	21
	2	82
	3	77
	5	211
	6	76
	7	13
2	3	109
	6	120
3	1	35
	8	215
4	1	14
	2	102
	3	22
	4	122
	6	42
	7	13
5	1	14
	3	33
	4	107
	5	84
	6	60
	7	16
	8	92

Table B-15. Test Set 15

Receiving Truck

Truck	Product	Quantity
1	1	50
	3	150
	4	150
2	1	167
	3	143
3	1	96
	2	96
	3	48
	4	110
4	3	310
5	1	85
	2	85
	3	53
	4	97
6	2	156
	3	156
	4	78

Shipping Truck

Truck	Product	Quantity
1	1	31
	2	58
	3	138
	4	198
2	2	93
	3	230
	4	40
3	1	61
	2	35
	3	91
	4	119
4	2	105
	3	401
5	1	306
	2	46
	4	78

Table B-16. Test Set 16

Receiving Truck

Truck	Product	Quantity
1	1	39
	2	53
	4	26
	5	132
2	3	350
3	2	121
	4	97
	5	60
	6	72
4	2	162
	3	65
	4	163
5	1	55
	4	240
	6	55

Shipping Truck

Truck	Product	Quantity
1	3	259
2	1	37
	3	26
	4	94
	5	34
	6	13
3	1	8
	2	131
	3	26
	4	19
	5	68
	6	40
4	1	24
	4	94
	5	21
	6	13
5	2	147
	4	188
6	1	25
	2	58
	3	104
	4	131
	5	69
	6	61

Table B-17. Test Set 17

Receiving Truck

Truck	Product	Quantity
1	1	90
	2	103
	4	77
2	1	112
	2	22
	3	56
	4	34
	5	56
3	5	180
	6	100
4	1	84
	2	14
	3	98
	4	28
	5	56
	6	28
	7	42

Shipping Truck

Truck	Product	Quantity
1	3	42
	5	203
	7	42
2	1	286
3	2	139
	4	139
	5	89
	6	128
4	3	112

Table B-18. Test Set 18

Receiving Truck

Truck	Product	Quantity
1	3	290
2	2	275
	4	55
3	6	320
4	2	270
5	1	29
	2	53
	3	35
	4	47
	5	29
	6	53
	7	54
6	1	59
	2	7
	4	59
	5	63
	6	59
	7	13

Shipping Truck

Truck	Product	Quantity
1	4	32
	6	288
2	4	35
	5	23
	6	103
3	1	88
	2	173
	3	217
	4	8
	5	7
	6	41
	7	5
4	2	259
5	2	173
	3	108
	4	8
	5	30
	7	6
6	4	78
	5	32
	7	56

Table B-19. Test Set 19

Receiving Truck

Truck	Product	Quantity
1	1	35
	2	35
	3	6
	4	60
	5	35
	6	53
	7	18
	8	29
	9	53
	10	6
2	2	138
	4	139
	5	23
3	2	31
	3	54
	4	46
	5	23
	6	38
	7	31
	8	38
	10	69
4	1	54
	4	122
	5	81
	9	54
	10	69
5	6	380

Shipping Truck

Truck	Product	Quantity
1	3	17
	5	27
	6	290
	7	10
	9	6
2	3	19
	5	27
	7	21
	9	48
	10	85
3	3	18
	5	81
	7	18
	9	39
	10	41
4	1	56
	4	321
	5	27
	8	67
	9	7
	10	9
5	1	33
	2	204
	3	6
	4	46
	6	181
	9	7
	10	9

Table B-20. Test Set 20

Receiving Truck

Truck	Product	Quantity
1	7	53
	8	238
	9	79
2	2	293
	9	97
3	1	38
	3	13
	4	125
	5	38
	7	38
	8	88
4	3	102
	5	128
	6	25
	8	25
5	1	57
	2	8
	3	49
	4	74
	5	74
	6	49
	7	16
	8	16
	9	17
6	1	64
	3	64
	4	25
	6	38
	7	89

Shipping Truck

Truck	Product	Quantity
1	8	204
2	1	44
	2	30
	3	228
	4	93
	5	70
	6	46
	7	25
	8	61
	9	69
3	2	91
	5	50
	7	48
4	1	88
	5	60
	6	66
	7	110
5	2	180
6	1	27
	4	131
	5	60
	7	13
	8	102
	9	124

APPENDIX C. OPTIMAL SOLUTIONS FOR THE *CASE 2* PROBLEM

Table C-1. Optimal Solution of Test Set 1 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	1	48
		4	72
1	3	2	4
1	4	2	27
1	5	2	5
		3	84
2	1	1	89
2	2	2	106
		3	18
2	5	2	21
		3	46
3	1	1	14
		4	15
3	2	3	15
3	4	1	61
		2	105
4	3	2	260

Minimum Number of Matching Pairs : 11

Table C-2. Optimal Solution of Test Set 2 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	1	48
		2	85
1	3	6	63
1	4	6	34
2	2	2	29
2	4	2	28
		4	47
		5	66
3	1	1	73
		3	32
		5	16
		6	10
3	4	1	7
		3	38
		5	54
4	1	4	24
		5	30
		6	13
4	4	1	18
		2	61
		4	19
		6	5
5	1	1	1
		3	43
		4	75
		5	26
5	2	1	75
		2	10

Minimum Number of Matching Pairs : 11

Table C-3. Optimal Solution of Test Set 3 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	2	28
		5	32
		6	24
		7	13
1	2	1	14
		3	55
		6	24
1	3	4	41
		5	64
		7	15
		8	28
2	2	1	116
		3	26
2	3	2	41
		4	50
		8	39
3	1	2	22
		8	102
3	2	3	64
3	3	2	16
		4	64
		8	12

Minimum Number of Matching Pairs : 8

Table C-4. Optimal Solution of Test Set 4 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	2	4
		3	6
		5	50
		6	22
		8	9
1	3	4	22
		7	39
		8	2
1	4	1	39
		2	2
		5	7
		8	28
2	2	4	46
		6	38
2	4	1	64
		4	25
2	5	1	7
		6	40
3	2	3	69
3	3	3	41
3	4	3	50
4	1	2	12
4	3	4	9
		5	6
4	4	2	11
		3	25
		4	36
		5	23
4	5	2	17
		5	21
5	1	8	42
5	3	5	14
		7	72
		8	16
5	5	1	86

Minimum Number of Matching Pairs : 16

Table C-5. Optimal Solution of Test Set 5 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	1	26
1	2	1	144
2	1	6	6
		7	14
		8	9
2	2	1	6
		5	38
2	3	2	6
		3	19
		4	50
		7	5
		8	47
3	1	1	49
		2	12
		3	59
		6	3
		7	37
		8	31
3	3	2	19
		3	1
		6	9
4	1	7	47
4	2	5	143
5	2	5	36
5	3	4	58
		7	72
		8	14

Minimum Number of Matching Pairs : 11

Table C-6. Optimal Solution of Test Set 6 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	1	22
1	2	5	40
1	3	3	32
		4	37
		5	33
1	4	2	22
		3	34
2	3	3	60
		4	64
2	4	2	100
		4	26
3	2	4	159
		5	43
3	3	4	58
4	2	5	50
4	3	1	3
4	4	1	8
		2	89
		3	22
		4	101
		5	17

Minimum Number of Matching Pairs : 11

Table C-7. Optimal Solution of Test Set 7 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	2	4
		3	28
1	3	1	14
		2	65
		5	69
2	1	1	27
		6	70
2	3	1	23
		4	40
3	2	2	190
4	1	1	23
		3	17
4	3	3	18
		5	92
4	4	3	80
5	1	3	19
		6	25
5	2	3	25
		5	62
		6	74
5	3	5	4
		6	11

Minimum Number of Matching Pairs : 11

Table C-8. Optimal Solution of Test Set 8 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	2	38
		3	75
		7	22
1	2	2	37
		7	12
1	3	1	61
		7	20
1	4	1	3
		7	17
1	5	7	25
2	1	3	38
2	3	3	131
2	5	3	131
3	2	2	21
		5	50
		6	12
3	3	2	19
		4	11
		5	10
3	4	1	37
		4	18
		5	5
		6	26
3	5	1	20
		2	32
		4	7
		6	12

Minimum Number of Matching Pairs : 12

Table C-9. Optimal Solution of Test Set 9 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	2	32
		4	45
		6	41
1	3	5	32
		8	32
1	4	3	18
2	1	1	58
		2	12
		5	10
		6	33
		7	6
2	2	2	10
		6	2
2	3	5	11
		8	46
2	4	2	30
		5	26
		7	6
3	1	1	187
		5	27
3	2	5	26
4	1	4	52
4	2	4	47
4	4	3	111

Minimum Number of Matching Pairs : 12

Table C-10. Optimal Solution of Test Set 10 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	4	38
		6	29
		8	7
1	2	5	19
		6	10
		7	12
1	3	2	41
		6	19
		9	58
1	4	1	64
		2	17
		3	19
		7	7
2	2	5	132
2	3	9	118
3	1	8	21
3	2	8	28
3	3	7	61
		9	53
3	4	2	49
		7	36
		9	92

Minimum Number of Matching Pairs : 10

Table C-11. Optimal Solution of Test Set 11 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	1	35
		2	25
		3	76
		4	25
		6	37
1	4	1	15
		3	47
2	1	1	18
		2	64
		4	32
		6	33
2	2	3	37
		4	14
		5	20
		6	4
2	4	1	55
		5	53
3	3	2	360
4	1	2	29
		3	14
		4	59
		5	19
4	4	3	104
		5	55
5	1	2	107
5	2	2	109
5	3	2	174

Minimum Number of Matching Pairs : 11

Table C-12. Optimal Solution of Test Set 12 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	8	32
1	3	2	46
		4	83
		6	18
		8	15
1	4	1	83
		3	83
2	1	7	340
3	2	2	62
		4	60
3	3	2	67
		4	37
		8	31
3	4	3	113
4	2	6	109
4	4	1	47
		3	94
5	2	5	10
		6	17
		8	43
5	3	5	51
		6	61
5	4	3	78
6	1	7	76
		8	14
6	2	2	50
		4	59
		5	76
		6	59
		8	11
6	4	1	25

Minimum Number of Matching Pairs : 15

Table C-13. Optimal Solution of Test Set 13 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	1	293
1	3	7	37
2	5	3	310
3	1	7	53
3	2	1	75
		2	32
3	3	6	19
		7	32
3	4	2	62
		5	57
4	2	8	13
4	3	6	38
		7	28
4	4	1	74
		5	52
		6	17
		7	8
4	5	3	15
4	6	2	74
		6	12
		7	9
		8	10
5	1	7	47
5	2	2	9
		4	42
		5	48
		6	24
5	4	5	36
5	5	3	19
5	6	2	19
		4	42
		6	4

Minimum Number of Matching Pairs : 17

Table C-14. Optimal Solution of Test Set 14 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	3	70
		5	118
1	2	3	8
1	5	3	33
		5	39
		8	32
2	2	6	120
2	3	8	186
2	5	6	16
		8	18
3	2	3	101
3	3	1	9
3	5	1	14
		4	68
		5	34
		6	44
4	1	1	21
		2	34
		7	3
4	4	1	11
		2	102
		4	56
		6	32
		7	13
4	5	4	39
		5	11
		7	16
		8	42
5	1	2	48
		3	7
		5	93
		6	76
		7	10
5	3	1	26
		8	29
5	4	1	3
		3	22
		4	66
		6	10

Minimum Number of Matching Pairs : 15

Table C-15. Optimal Solution of Test Set 15 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	3	85
		4	101
1	3	1	50
		3	65
		4	49
2	2	3	143
2	5	1	167
3	2	2	77
		3	48
		4	32
3	3	1	11
3	5	1	85
		2	19
		4	78
4	4	3	310
5	1	1	31
		2	58
		3	53
		4	97
5	5	1	54
		2	27
6	2	2	16
		3	39
		4	8
6	3	2	35
		3	26
		4	70
6	4	2	105
		3	91

Minimum Number of Matching Pairs : 13

Table C-16. Optimal Solution of Test Set 16 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	1	6
		5	34
1	3	1	8
		2	53
		5	68
1	6	1	25
		4	26
		5	30
2	1	3	259
2	2	3	26
2	6	3	65
3	4	5	21
		6	11
3	5	2	121
		4	97
3	6	5	39
		6	61
4	3	2	78
		3	26
4	5	2	26
		4	58
4	6	2	58
		3	39
		4	105
5	2	1	31
		4	94
		6	13
5	3	4	19
		6	40
5	4	1	24
		4	94
		6	2
5	5	4	33

Minimum Number of Matching Pairs : 16

Table C-17. Optimal Solution of Test Set 17 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	2	1	90
1	3	2	103
		4	77
2	2	1	112
2	3	2	22
		4	34
		5	56
2	4	3	56
3	1	5	180
3	3	6	100
4	1	3	42
		5	23
		7	42
4	2	1	84
4	3	2	14
		4	28
		5	33
		6	28
4	4	3	56

Minimum Number of Matching Pairs : 11

Table C-18. Optimal Solution of Test Set 18 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	3	3	182
1	5	3	108
2	4	2	259
2	5	2	16
2	6	4	55
3	1	6	270
3	2	6	50
4	3	2	120
4	5	2	150
5	2	4	35
		5	23
		6	53
5	3	1	29
		2	53
		3	35
		4	8
		7	5
5	6	4	4
		5	6
		7	49
6	1	4	32
		6	18
6	3	1	59
		5	7
		6	41
6	5	2	7
		4	8
		5	30
		7	6
6	6	4	19
		5	26
		7	7

Minimum Number of Matching Pairs : 16

Table C-19. Optimal Solution of Test Set 19 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	5	27
		9	6
1	3	5	8
		7	18
		9	39
1	4	1	2
		4	60
		8	29
		9	1
1	5	1	33
		2	35
		3	6
		6	53
		9	7
		10	6
2	4	4	139
		5	23
2	5	2	138
3	1	3	17
		6	38
		7	10
3	2	3	19
		7	21
		10	16
3	3	3	18
		5	23
		10	41
3	4	8	38
		10	9
3	5	2	31
		4	46
		10	3
4	2	5	27
		9	48
		10	69
4	3	5	50
4	4	1	54
		4	122
		5	4
		9	6
5	1	6	252
5	5	6	128

Minimum Number of Matching Pairs : 16

Table C-20. Optimal Solution of Test Set 20 for the *Case 2* Problem

Receiving Truck	Shipping Truck	Product	Unit
1	1	8	204
1	3	7	48
1	6	7	5
		8	34
		9	79
2	2	2	22
		9	52
2	3	2	91
2	5	2	180
2	6	9	45
3	2	1	38
		3	13
		7	25
		8	20
3	4	7	5
3	6	4	125
		5	38
		7	8
		8	68
4	2	3	102
		5	68
		8	25
4	4	5	60
		6	25
5	2	1	6
		2	8
		3	49
		4	68
		5	2
		6	46
		8	16
		9	17
5	3	5	50
5	4	1	24
		6	3
		7	16
5	6	1	27
		4	6
		5	22
6	2	3	64
		4	25
6	4	1	64
		6	38
		7	89

Minimum Number of Matching Pairs : 18

APPENDIX D. OPTIMAL SOLUTIONS FOUND AFTER MINIMIZING MEAN FLOW TIME FOR THE *CASE 2* PROBLEM

The complete enumeration method was used to find the optimal solution for the *Case 2* problem. The optimal solution was only found in eleven test problems among twenty test problems because of computational time. The eleven test problems are: 1, 2, 3, 5, 6, 7, 8, 9, 10, 11 and 17.

TEST SET 1

BEST RECEIVING TRUCK SEQUENCE =
2 3 3 1 3 1 2 2 1 1 4
BEST SHIPPING TRUCK SEQUENCE =
2 2 4 4 1 1 1 5 5 3 3

BEST MEAM FLOW TIME = 466.555556

WORST RECEIVING TRUCK SEQUENCE =
1 3 2 2 3 1 4 1 2 1 3
WORST SHIPPING TRUCK SEQUENCE =
3 1 5 2 4 1 3 5 1 4 2

WORST MEAM FLOW TIME = 1263.000000

AVERAGE MEAM FLOW TIME = 856.703704

The Running Time is : 1023.44000000

TEST SET 2

BEST RECEIVING TRUCK SEQUENCE =
1 1 1 2 2 5 5 4 4 3 3
BEST SHIPPING TRUCK SEQUENCE =
3 2 4 2 4 2 1 1 4 4 1

BEST MEAM FLOW TIME = 482.000000

WORST RECEIVING TRUCK SEQUENCE =
1 5 3 4 2 1 5 1 3 4 2
WORST SHIPPING TRUCK SEQUENCE =
4 2 1 4 4 2 1 3 4 1 2

WORST MEAM FLOW TIME = 1193.444444

AVERAGE MEAM FLOW TIME = 864.829630

The Running Time is : 1013.99400000

TEST SET 3

BEST RECEIVING TRUCK SEQUENCE =
2 2 1 1 3 3 1 3
BEST SHIPPING TRUCK SEQUENCE =
2 3 2 3 2 3 1 1

BEST MEAM FLOW TIME = 623.833333

WORST RECEIVING TRUCK SEQUENCE =
1 3 2 1 3 2 1 3
WORST SHIPPING TRUCK SEQUENCE =
2 1 3 3 2 2 1 3

WORST MEAM FLOW TIME = 1068.000000

AVERAGE MEAM FLOW TIME = 857.583333

The Running Time is : 1.25100000

TEST SET 5

BEST RECEIVING TRUCK SEQUENCE =
1 1 4 4 3 2 2 3 2 5 5
BEST SHIPPING TRUCK SEQUENCE =
2 1 1 2 1 1 2 3 3 2 3

BEST MEAM FLOW TIME = 565.625000

WORST RECEIVING TRUCK SEQUENCE =
1 2 5 4 3 2 1 4 5 2 3
WORST SHIPPING TRUCK SEQUENCE =
1 2 3 1 1 3 2 2 2 1 3

WORST MEAM FLOW TIME = 1319.750000

AVERAGE MEAM FLOW TIME = 949.083333

The Running Time is : 990.17400000

TEST SET 6

BEST RECEIVING TRUCK SEQUENCE =
2 2 4 4 1 1 1 1 4 3 3
BEST SHIPPING TRUCK SEQUENCE =
4 3 3 4 4 1 3 2 2 3 2

BEST MEAM FLOW TIME = 600.000000

WORST RECEIVING TRUCK SEQUENCE =
4 1 3 2 1 1 4 3 2 1 4
WORST SHIPPING TRUCK SEQUENCE =
2 4 3 3 1 3 4 2 4 2 3

WORST MEAM FLOW TIME = 1290.625000

AVERAGE MEAM FLOW TIME = 941.162500

The Running Time is : 992.35800000

TEST SET 7

BEST RECEIVING TRUCK SEQUENCE =
3 5 1 1 5 5 2 2 4 4 4
BEST SHIPPING TRUCK SEQUENCE =
2 2 2 3 3 1 1 3 1 3 4

BEST MEAM FLOW TIME = 423.222222

WORST RECEIVING TRUCK SEQUENCE =
4 2 5 1 3 4 5 4 2 1 5
WORST SHIPPING TRUCK SEQUENCE =
1 3 2 3 2 3 1 4 1 2 3

WORST MEAM FLOW TIME = 1134.666667

AVERAGE MEAM FLOW TIME = 808.822222

The Running Time is : 1082.59400000

TEST SET 8

BEST RECEIVING TRUCK SEQUENCE =
1 2 1 2 2 1 3 3 1 3 1 3
BEST SHIPPING TRUCK SEQUENCE =
1 1 3 3 5 5 3 5 2 2 4 4

BEST MEAM FLOW TIME = 590.000000

WORST RECEIVING TRUCK SEQUENCE =
1 3 2 1 1 2 3 3 1 3 2 1
WORST SHIPPING TRUCK SEQUENCE =
4 3 5 2 1 3 5 4 3 2 1 5

WORST MEAM FLOW TIME = 1328.750000

AVERAGE MEAM FLOW TIME = 987.958333

The Running Time is : 13005.20200000

TEST SET 9

BEST RECEIVING TRUCK SEQUENCE =
1 2 1 1 2 2 4 4 4 2 3 3
BEST SHIPPING TRUCK SEQUENCE =
3 3 2 4 2 4 4 2 1 1 2 1

BEST MEAM FLOW TIME = 597.250000

WORST RECEIVING TRUCK SEQUENCE =
1 3 4 2 1 2 2 4 3 4 1 2
WORST SHIPPING TRUCK SEQUENCE =
4 2 1 3 2 1 4 2 1 4 3 2

WORST MEAM FLOW TIME = 1391.750000

AVERAGE MEAM FLOW TIME = 1020.608333

The Running Time is : 28081.29400000

TEST SET 10

BEST RECEIVING TRUCK SEQUENCE =
2 2 1 1 3 3 1 3 1 3
BEST SHIPPING TRUCK SEQUENCE =
2 3 2 3 2 3 1 1 4 4

BEST MEAM FLOW TIME = 594.142857

WORST RECEIVING TRUCK SEQUENCE =
3 1 2 1 1 3 3 2 1 3
WORST SHIPPING TRUCK SEQUENCE =
1 4 2 3 2 3 4 3 1 2

WORST MEAM FLOW TIME = 1263.857143

AVERAGE MEAM FLOW TIME = 940.404762

The Running Time is : 75.00400000

TEST SET 11

BEST RECEIVING TRUCK SEQUENCE =
3 5 5 5 1 1 2 2 2 4 4
BEST SHIPPING TRUCK SEQUENCE =
3 3 2 1 2 4 2 1 4 1 4

BEST MEAM FLOW TIME = 649.444444

WORST RECEIVING TRUCK SEQUENCE =
1 2 5 4 3 2 5 1 5 4 2
WORST SHIPPING TRUCK SEQUENCE =
4 2 1 4 3 1 2 2 3 1 4

WORST MEAM FLOW TIME = 1657.777778

AVERAGE MEAM FLOW TIME = 1199.888889

The Running Time is : 1011.76400000

TEST SET 17

BEST RECEIVING TRUCK SEQUENCE =
1 1 2 2 2 4 4 4 3 4 3
BEST SHIPPING TRUCK SEQUENCE =
3 2 2 3 4 4 2 3 3 1 1

BEST MEAM FLOW TIME = 623.625000

WORST RECEIVING TRUCK SEQUENCE =
2 4 3 4 1 2 4 3 2 1 4
WORST SHIPPING TRUCK SEQUENCE =
4 2 3 1 3 2 3 1 3 2 4

WORST MEAM FLOW TIME = 1458.250000

AVERAGE MEAM FLOW TIME = 1075.375000

The Running Time is : 1002.52700000

APPENDIX E. TABU SEARCH SOLUTIONS FOUND AFTER MINIMIZING MEAN FLOW TIME FOR THE *CASE 2* PROBLEM

The tabu search method was used to find the optimal solution. Using the tabu search, the solutions were found in all twenty test problems.

TEST SET 1

RECEIVING TRUCK SEQUENCE =
2 3 3 3 2 1 1 2 1 1 4
SHIPPING TRUCK SEQUENCE =
2 2 4 1 1 4 1 5 5 3 3

MEAM FLOW TIME = 466.555556

The Running Time is : 5.82300000

TEST SET 2

RECEIVING TRUCK SEQUENCE =
1 1 1 2 2 5 5 4 4 3 3
SHIPPING TRUCK SEQUENCE =
3 2 4 4 2 2 1 4 1 4 1

MEAM FLOW TIME = 482.000000

The Running Time is : 5.82300000

TEST SET 3

RECEIVING TRUCK SEQUENCE =
3 1 3 3 1 1 2 2
SHIPPING TRUCK SEQUENCE =
1 1 3 2 2 3 3 2

MEAM FLOW TIME = 623.833333

The Running Time is : 4.45000000

TEST SET 4

RECEIVING TRUCK SEQUENCE =
1 1 1 3 3 3 2 2 2 4 4 4 4 5 5 5
SHIPPING TRUCK SEQUENCE =
2 4 3 2 4 3 2 4 5 4 3 5 1 1 5 3

MEAM FLOW TIME = 668.500000

The Running Time is : 10.17900000

TEST SET 5

RECEIVING TRUCK SEQUENCE =
1 1 4 4 3 2 2 3 2 5 5
SHIPPING TRUCK SEQUENCE =
2 1 1 2 1 1 2 3 3 2 3

MEAM FLOW TIME = 565.625000

The Running Time is : 6.96600000

TEST SET 6

RECEIVING TRUCK SEQUENCE =
3 3 4 4 1 1 1 1 4 2 2
SHIPPING TRUCK SEQUENCE =
2 3 3 2 2 3 1 4 4 3 4

MEAM FLOW TIME = 600.000000

The Running Time is : 6.12100000

TEST SET 7

RECEIVING TRUCK SEQUENCE =
4 4 4 2 2 5 5 1 1 5 3
SHIPPING TRUCK SEQUENCE =
4 3 1 1 3 1 3 3 2 2 2

MEAM FLOW TIME = 423.222222

The Running Time is : 6.08400000

TEST SET 8

RECEIVING TRUCK SEQUENCE =
1 2 1 2 2 1 3 3 1 3 1 3
SHIPPING TRUCK SEQUENCE =
1 1 3 3 5 5 3 5 2 2 4 4

MEAM FLOW TIME = 590.000000

The Running Time is : 9.19100000

TEST SET 9

RECEIVING TRUCK SEQUENCE =
3 3 2 4 4 4 2 1 1 2 2 1
SHIPPING TRUCK SEQUENCE =
1 2 1 1 2 4 4 4 2 2 3 3

MEAM FLOW TIME = 597.250000

The Running Time is : 7.39600000

TEST SET 10

RECEIVING TRUCK SEQUENCE =
2 2 1 1 3 3 3 1 1 3
SHIPPING TRUCK SEQUENCE =
2 3 3 2 2 3 1 1 4 4

MEAM FLOW TIME = 594.142857

The Running Time is : 5.45500000

TEST SET 11

RECEIVING TRUCK SEQUENCE =
4 4 2 2 2 1 1 5 5 5 3
SHIPPING TRUCK SEQUENCE =
4 1 1 4 2 4 2 1 2 3 3

MEAM FLOW TIME = 649.444444

The Running Time is : 6.60600000

TEST SET 12

RECEIVING TRUCK SEQUENCE =
2 6 1 6 6 1 1 5 5 5 3 3 3 4 4
SHIPPING TRUCK SEQUENCE =
1 1 1 2 4 4 3 3 4 2 3 2 4 2 4

MEAM FLOW TIME = 871.100000

The Running Time is : 12.82700000

TEST SET 13

RECEIVING TRUCK SEQUENCE =
1 1 3 3 4 3 4 3 5 5 4 5 4 5 4 5 2
SHIPPING TRUCK SEQUENCE =
2 3 2 3 3 4 4 1 4 1 2 2 6 6 5 5 5

MEAM FLOW TIME = 740.363636

The Running Time is : 17.22200000

TEST SET 14

RECEIVING TRUCK SEQUENCE =
2 2 2 3 3 3 5 1 1 1 5 4 4 5 4
SHIPPING TRUCK SEQUENCE =
3 2 5 3 2 5 3 2 5 1 1 1 5 4 4

MEAM FLOW TIME = 832.800000

The Running Time is : 9.43700000

TEST SET 15

RECEIVING TRUCK SEQUENCE =
5 1 1 5 3 3 2 2 3 6 6 6 4
SHIPPING TRUCK SEQUENCE =
1 1 3 5 3 5 5 2 2 2 3 4 4

MEAM FLOW TIME = 679.454545

The Running Time is : 8.15300000

TEST SET 16

RECEIVING TRUCK SEQUENCE =
5 3 5 3 4 3 4 4 5 1 1 1 5 2 2 2
SHIPPING TRUCK SEQUENCE =
4 4 5 5 5 6 6 3 3 3 6 2 2 2 6 1

MEAM FLOW TIME = 728.000000

The Running Time is : 13.52800000

TEST SET 17

RECEIVING TRUCK SEQUENCE =
3 4 3 4 4 4 2 2 2 1 1
SHIPPING TRUCK SEQUENCE =
1 1 3 3 2 4 4 3 2 2 3

MEAM FLOW TIME = 623.625000

The Running Time is : 6.48500000

TEST SET 18

RECEIVING TRUCK SEQUENCE =
2 2 2 4 4 1 1 6 6 6 5 5 5 3 6 3
SHIPPING TRUCK SEQUENCE =
4 6 5 5 3 5 3 5 3 6 6 3 2 2 1 1

MEAM FLOW TIME = 674.666667

The Running Time is : 18.13900000

TEST SET 19

RECEIVING TRUCK SEQUENCE =
4 3 4 4 3 3 1 1 2 2 3 1 1 3 5 5
SHIPPING TRUCK SEQUENCE =
2 2 4 3 4 3 3 4 4 5 5 5 1 1 5 1

MEAM FLOW TIME = 828.700000

The Running Time is : 11.84800000

TEST SET 20

RECEIVING TRUCK SEQUENCE =
4 4 6 6 3 3 3 5 5 2 2 5 5 2 2 1 1 1
SHIPPING TRUCK SEQUENCE =
2 4 2 4 2 4 6 4 2 2 6 6 3 3 5 3 6 1

MEAM FLOW TIME = 785.750000

The Running Time is : 15.04100000

APPENDIX F. BEST SOLUTIONS GENERATED BY HEURISTIC ALGORITHM FOR THE *CASE 3* PROBLEM WHERE TRUCK CHANGE TIME IS 75

TEST SET 1

Receiving Truck Sequence:
4 1 3 2 4 3

Shipping Truck Sequence:
5 1 3 2 4

Product Routing [1]: Receiving Truck:
4 4 4 1 3 3 2 2 2 4 4 3

Product Routing [2]: Shipping Truck:
5 1 3 1 3 2 3 2 4 2 4 4

Product Routing [3]: Number of Total Products transferred:
139 4 28 260 120 10 90 105 15 78 31 110

TEST SET 2

Receiving Truck Sequence:
1 2 5 3 4 2 5

Shipping Truck Sequence:
3 2 1 4

Product Routing [1]: Receiving Truck:
1 1 1 2 5 5 3 3 4 4 2 5

Product Routing [2]: Shipping Truck:
3 2 4 2 2 1 1 4 1 4 4 4

Product Routing [3]: Number of Total Products transferred:
63 133 34 29 85 61 224 6 58 112 141 84

TEST SET 3

Receiving Truck Sequence:
3 1 3 2 1

Shipping Truck Sequence:
1 3 2

Product Routing [1]: Receiving Truck:
3 1 1 3 3 2 2 1

Product Routing [2]: Shipping Truck:
1 1 3 3 2 3 2 2

Product Routing [3]: Number of Total Products transferred:
96 125 120 120 64 130 142 93

TEST SET 4

Receiving Truck Sequence:
2 4 1 5 4 3 5 1 2 3

Shipping Truck Sequence:
5 1 4 3 2

Product Routing [1]: Receiving Truck:
2 4 4 1 1 1 5 5 4 4 3 3 5 5 1 1 2 2 3

Product Routing [2]: Shipping Truck:
5 5 1 5 1 4 1 4 4 2 4 3 3 2 3 2 3 2 2

Product Routing [3]: Number of Total Products transferred:
111 38 6 22 45 33 3 115 112 4 50 35 103 9 58 72 25 84 75

TEST SET 5

Receiving Truck Sequence:
5 3 2 1 4 3

Shipping Truck Sequence:
3 2 1

Product Routing [1]: Receiving Truck:
5 5 5 3 2 2 2 1 1 4 4 3

Product Routing [2]: Shipping Truck:
3 2 1 3 3 2 1 2 1 2 1 1

Product Routing [3]: Number of Total Products transferred:
121 36 23 23 156 38 6 150 20 143 47 197

TEST SET 6

Receiving Truck Sequence:
1 2 4 3 1 2

Shipping Truck Sequence:
1 4 2 3

Product Routing [1]: Receiving Truck:
1 1 1 2 4 4 4 3 3 1 2

Product Routing [2]: Shipping Truck:
1 4 2 4 4 2 3 2 3 3 3

Product Routing [3]: Number of Total Products transferred:
22 22 40 160 237 50 3 202 58 136 90

TEST SET 7

Receiving Truck Sequence:
4 2 5 3 1 4

Shipping Truck Sequence:
4 1 2 3

Product Routing [1]: Receiving Truck:
4 4 2 2 5 5 5 3 1 1 4

Product Routing [2]: Shipping Truck:
4 2 1 3 1 2 3 2 2 3 3

Product Routing [3]: Number of Total Products transferred:
80 35 120 40 61 144 15 190 14 166 115

TEST SET 8

Receiving Truck Sequence:
2 1 3 2 1

Shipping Truck Sequence:
1 2 4 3 5

Product Routing [1]: Receiving Truck:
2 1 1 1 1 3 3 3 3 2 2 1

Product Routing [2]: Shipping Truck:
1 1 2 4 3 2 4 3 5 3 5 5

Product Routing [3]: Number of Total Products transferred:
38 135 12 17 69 120 89 52 19 131 131 77

TEST SET 9

Receiving Truck Sequence:
2 1 4 3 2 4

Shipping Truck Sequence:
3 2 1 4

Product Routing [1]: Receiving Truck:
2 2 1 1 1 4 4 3 3 2 2 4

Product Routing [2]: Shipping Truck:
3 2 3 2 4 2 1 1 4 1 4 4

Product Routing [3]: Number of Total Products transferred:
89 12 32 144 24 47 52 224 16 109 40 111

TEST SET 10

Receiving Truck Sequence:
3 1 3 2

Shipping Truck Sequence:
4 1 3 2

Product Routing [1]: Receiving Truck:
3 3 1 1 1 1 3 3 2 2

Product Routing [2]: Shipping Truck:
4 1 4 1 3 2 3 2 3 2

Product Routing [3]: Number of Total Products transferred:
66 21 218 74 19 29 213 40 118 132

TEST SET 11

Receiving Truck Sequence:
5 3 4 2 1 3 2

Shipping Truck Sequence:
3 4 2 1

Product Routing [1]: Receiving Truck:
5 3 4 4 4 2 2 1 1 3 3 2

Product Routing [2]: Shipping Truck:
3 3 4 2 1 4 2 2 1 2 1 1

Product Routing [3]: Number of Total Products transferred:
390 144 149 63 68 180 68 235 25 16 200 82

TEST SET 12

Receiving Truck Sequence:
2 6 3 5 6 4 1 4 3 5

Shipping Truck Sequence:
1 3 2 4

Product Routing [1]: Receiving Truck:
2 6 6 3 3 3 5 5 6 6 4 1 1 4 3 5

Product Routing [2]: Shipping Truck:
1 1 3 1 3 2 3 2 2 4 2 2 4 4 4 4

Product Routing [3]: Number of Total Products transferred:
340 101 24 21 213 23 172 10 220 25 109 194 166 141 113 78

TEST SET 13

Receiving Truck Sequence:
3 4 5 2 3 5 1 4

Shipping Truck Sequence:
3 6 5 4 1 2

Product Routing [1]: Receiving Truck:
3 4 4 4 4 4 5 5 2 3 3 5 5 5 1 1 4

Product Routing [2]: Shipping Truck:
3 3 6 5 4 1 6 5 5 4 2 4 1 2 1 2 2

Product Routing [3]: Number of Total Products transferred:
104 50 109 15 4 24 61 19 310 193 33 109 39 62 37 293 148

TEST SET 14

Receiving Truck Sequence:
3 4 5 1 2 3 5 1

Shipping Truck Sequence:
4 1 3 2 5

Product Routing [1]: Receiving Truck:
3 3 4 4 4 4 5 1 1 2 2 2 3 3 5 1

Product Routing [2]: Shipping Truck:
4 3 4 1 3 5 1 1 2 3 2 5 2 5 5 5

Product Routing [3]: Number of Total Products transferred:
59 17 256 37 29 58 277 166 30 204 120 16 79 115 113 104

TEST SET 15

Receiving Truck Sequence:
4 6 5 6 1 3 2 3

Shipping Truck Sequence:
4 3 1 5 2

Product Routing [1]: Receiving Truck:
4 6 6 5 5 5 5 6 6 1 1 1 3 2 2 3

Product Routing [2]: Shipping Truck:
4 4 3 3 1 5 2 1 2 1 5 2 5 5 2 2

Product Routing [3]: Number of Total Products transferred:
310 196 60 246 7 24 43 99 35 319 19 12 220 167 143 130

TEST SET 16

Receiving Truck Sequence:
2 4 1 3 5 2 3 1 5 4

Shipping Truck Sequence:
1 5 4 2 3 6

Product Routing [1]: Receiving Truck:
2 4 1 1 3 3 5 5 2 2 2 3 3 1 1 5 4

Product Routing [2]: Shipping Truck:
1 5 5 4 4 2 2 3 2 3 6 3 6 3 6 6 6

Product Routing [3]: Number of Total Products transferred:
259 310 25 24 128 34 144 15 26 26 39 169 19 82 119 191 80

TEST SET 17

Receiving Truck Sequence:
3 4 2 4 1 2 3

Shipping Truck Sequence:
1 4 2 3

Product Routing [1]: Receiving Truck:
3 4 4 2 2 4 4 1 1 2 3

Product Routing [2]: Shipping Truck:
1 1 4 4 2 2 3 2 3 3 3

Product Routing [3]: Number of Total Products transferred:
147 140 56 56 112 84 70 90 180 112 133

TEST SET 18

Receiving Truck Sequence:
2 3 5 6 4 1 6 5 1 4 6

Shipping Truck Sequence:
4 1 6 5 2 3

Product Routing [1]: Receiving Truck:
2 2 2 2 3 3 5 6 6 4 1 6 5 5 1 4 6

Product Routing [2]: Shipping Truck:
4 1 2 3 1 3 6 6 5 5 5 2 2 3 3 3 3

Product Routing [3]: Number of Total Products transferred:
259 32 15 24 288 32 130 36 44 173 108 102 44 126 182 97 78

TEST SET 19

Receiving Truck Sequence:
4 3 5 1 4 3 2 1 5 3

Shipping Truck Sequence:
2 1 3 4 5

Product Routing [1]: Receiving Truck:
4 3 3 3 5 1 1 1 4 4 3 2 2 1 5 3

Product Routing [2]: Shipping Truck:
2 2 1 3 1 1 3 4 3 4 4 4 5 5 5 5

Product Routing [3]: Number of Total Products transferred:
144 56 50 47 290 10 100 46 50 186 93 162 138 174 90 84

TEST SET 20

Receiving Truck Sequence:
1 2 6 5 2 3 1 4 5 6

Shipping Truck Sequence:
1 5 4 3 6 2

Product Routing [1]: Receiving Truck:
1 1 1 2 6 6 5 5 2 2 2 3 3 3 1 1 4 4 5 6

Product Routing [2]: Shipping Truck:
1 4 3 5 4 6 4 3 3 6 2 3 6 2 6 2 6 2 2 2

Product Routing [3]: Number of Total Products transferred:
204 5 10 180 191 6 128 12 91 45 74 76 240 24 106 45 60 220 220 83

APPENDIX G. BEST SOLUTIONS GENERATED BY HEURISTIC ALGORITHM FOR THE *CASE 3* PROBLEM WHERE TRUCK CHANGE TIME IS 15

TEST SET 1

Receiving Truck Sequence:
4 1 3 4 2 4 3 2

Shipping Truck Sequence:
5 1 3 2 4

Product Routing [1]: Receiving Truck:
4 4 1 3 3 4 2 2 4 4 3 2

Product Routing [2]: Shipping Truck:
5 1 1 3 2 3 3 2 2 4 4 4

Product Routing [3]: Number of Total Products transferred:
139 4 260 120 10 28 90 105 78 31 110 15

TEST SET 2

Receiving Truck Sequence:
1 2 5 3 4 2 5 1

Shipping Truck Sequence:
3 2 1 4

Product Routing [1]: Receiving Truck:
1 1 2 5 5 3 3 4 4 2 5 1

Product Routing [2]: Shipping Truck:
3 2 2 2 1 1 4 1 4 4 4 4

Product Routing [3]: Number of Total Products transferred:
63 133 29 85 61 224 6 58 112 141 84 34

TEST SET 3

Receiving Truck Sequence:
3 1 3 2 1 3

Shipping Truck Sequence:
1 3 2

Product Routing [1]: Receiving Truck:
3 1 1 3 2 2 1 3

Product Routing [2]: Shipping Truck:
1 1 3 3 3 2 2 2

Product Routing [3]: Number of Total Products transferred:
96 125 120 120 130 142 93 64

TEST SET 4

Receiving Truck Sequence:
2 4 1 5 4 3 1 5 3 2 3 1

Shipping Truck Sequence:
5 1 4 3 2

Product Routing [1]: Receiving Truck:
2 4 4 1 1 5 5 4 4 3 1 1 5 5 3 2 2 3 1

Product Routing [2]: Shipping Truck:
5 5 1 5 1 1 4 4 2 4 4 3 3 2 3 3 2 2 2

Product Routing [3]: Number of Total Products transferred:
111 38 6 22 45 3 115 112 4 50 33 58 103 9 35 25 84 75 72

TEST SET 5

Receiving Truck Sequence:
5 3 2 1 5 4 3 5 1

Shipping Truck Sequence:
3 2 1

Product Routing [1]: Receiving Truck:
5 3 2 2 2 1 5 4 4 3 5 1

Product Routing [2]: Shipping Truck:
3 3 3 2 1 2 2 2 1 1 1 1

Product Routing [3]: Number of Total Products transferred:
121 23 156 38 6 150 36 143 47 197 23 20

TEST SET 6

Receiving Truck Sequence:
1 2 4 3 1 2 3

Shipping Truck Sequence:
1 4 2 3

Product Routing [1]: Receiving Truck:
1 1 2 4 4 4 3 1 1 2 3

Product Routing [2]: Shipping Truck:
1 4 4 4 2 3 2 2 3 3 3

Product Routing [3]: Number of Total Products transferred:
22 22 160 237 50 3 202 40 136 90 58

TEST SET 7

Receiving Truck Sequence:
4 2 5 3 4 1 4 5 2

Shipping Truck Sequence:
4 1 2 3

Product Routing [1]: Receiving Truck:
4 2 5 5 3 4 1 1 4 5 2

Product Routing [2]: Shipping Truck:
4 1 1 2 2 2 2 3 3 3 3

Product Routing [3]: Number of Total Products transferred:
80 120 61 144 190 35 14 166 115 15 40

TEST SET 8

Receiving Truck Sequence:
2 1 3 1 3 2 1 3

Shipping Truck Sequence:
1 2 4 3 5

Product Routing [1]: Receiving Truck:
2 1 1 3 3 1 1 3 2 2 1 3

Product Routing [2]: Shipping Truck:
1 1 2 2 4 4 3 3 3 5 5 5

Product Routing [3]: Number of Total Products transferred:
38 135 12 120 89 17 69 52 131 131 77 19

TEST SET 9

Receiving Truck Sequence:
2 3 4 1 3 2 4 1

Shipping Truck Sequence:
1 2 3 4

Product Routing [1]: Receiving Truck:
2 2 3 4 4 1 1 3 3 2 2 4 1

Product Routing [2]: Shipping Truck:
1 2 1 1 2 2 3 3 4 3 4 4 4

Product Routing [3]: Number of Total Products transferred:
146 12 187 52 47 144 32 43 10 46 46 111 24

TEST SET 10

Receiving Truck Sequence:
3 1 3 1 2 3 1

Shipping Truck Sequence:
4 1 3 2

Product Routing [1]: Receiving Truck:
3 1 1 3 3 1 2 2 3 1

Product Routing [2]: Shipping Truck:
4 4 1 1 3 3 3 2 2 2

Product Routing [3]: Number of Total Products transferred:
66 218 74 21 213 19 118 132 40 29

TEST SET 11

Receiving Truck Sequence:
5 3 4 2 3 1 2 3 4

Shipping Truck Sequence:
3 4 2 1

Product Routing [1]: Receiving Truck:
5 3 4 2 2 3 1 1 2 3 4

Product Routing [2]: Shipping Truck:
3 3 4 4 2 2 2 1 1 1 1

Product Routing [3]: Number of Total Products transferred:
390 144 192 137 38 109 235 25 155 107 88

TEST SET 12

Receiving Truck Sequence:
2 6 3 5 6 3 4 1 4 6 3 5

Shipping Truck Sequence:
1 3 2 4

Product Routing [1]: Receiving Truck:
2 6 3 3 5 5 6 6 3 4 1 1 4 6 3 5

Product Routing [2]: Shipping Truck:
1 1 1 3 3 2 3 2 2 2 2 4 4 4 4 4

Product Routing [3]: Number of Total Products transferred:
340 101 21 213 172 10 24 220 23 109 194 166 141 25 113 78

TEST SET 13

Receiving Truck Sequence:
3 4 5 2 4 3 5 4 1 4 5 3

Shipping Truck Sequence:
3 6 5 4 1 2

Product Routing [1]: Receiving Truck:
3 4 4 5 5 2 4 4 3 5 5 4 1 1 4 5 3

Product Routing [2]: Shipping Truck:
3 3 6 6 5 5 5 4 4 4 1 1 1 2 2 2 2

Product Routing [3]: Number of Total Products transferred:
104 50 109 61 19 310 15 4 193 109 39 24 37 293 148 62 33

TEST SET 14

Receiving Truck Sequence:
5 4 1 2 4 1 3 5 4 1 2

Shipping Truck Sequence:
4 2 3 1 5

Product Routing [1]: Receiving Truck:
5 5 5 4 1 2 2 4 4 1 3 3 5 4 1 2

Product Routing [2]: Shipping Truck:
4 3 1 4 2 2 3 3 1 1 1 5 5 5 5 5

Product Routing [3]: Number of Total Products transferred:
202 3 9 113 109 120 204 43 138 157 176 94 176 86 34 16

TEST SET 15

Receiving Truck Sequence:
4 6 5 6 1 3 5 2 3 6 5

Shipping Truck Sequence:
4 3 1 5 2

Product Routing [1]: Receiving Truck:
4 6 6 5 5 6 1 1 1 3 5 2 2 3 6 5

Product Routing [2]: Shipping Truck:
4 4 3 3 1 1 1 5 2 5 5 5 2 2 2 2

Product Routing [3]: Number of Total Products transferred:
310 196 60 246 7 99 319 19 12 220 24 167 143 130 35 43

TEST SET 16

Receiving Truck Sequence:
2 4 1 3 5 2 3 1 5 1 4 2 3

Shipping Truck Sequence:
1 5 4 2 3 6

Product Routing [1]: Receiving Truck:
2 4 1 1 3 3 5 2 2 3 1 5 5 1 4 2 3

Product Routing [2]: Shipping Truck:
1 5 5 4 4 2 2 2 3 3 3 3 6 6 6 6 6

Product Routing [3]: Number of Total Products transferred:
259 310 25 24 128 34 144 26 26 169 82 15 191 119 80 39 19

TEST SET 17

Receiving Truck Sequence:
3 4 2 4 1 2 4 3

Shipping Truck Sequence:
1 4 2 3

Product Routing [1]: Receiving Truck:
3 4 4 2 2 4 1 1 2 4 3

Product Routing [2]: Shipping Truck:
1 1 4 4 2 2 2 3 3 3 3

Product Routing [3]: Number of Total Products transferred:
180 107 56 56 112 84 90 180 112 103 100

TEST SET 18

Receiving Truck Sequence:
4 3 6 1 2 5 6 3 1 5 2 6

Shipping Truck Sequence:
4 1 5 6 2 3

Product Routing [1]: Receiving Truck:
4 4 3 6 6 6 1 2 2 2 5 5 6 3 1 5 2 6

Product Routing [2]: Shipping Truck:
4 3 1 1 5 6 5 5 6 2 6 2 2 2 3 3 3 3

Product Routing [3]: Number of Total Products transferred:
259 11 288 32 36 5 108 181 31 8 130 12 109 32 182 158 110 78

TEST SET 19

Receiving Truck Sequence:
4 3 5 4 1 4 3 2 1 3 5

Shipping Truck Sequence:
2 1 3 4 5

Product Routing [1]: Receiving Truck:
4 3 3 3 5 4 4 1 1 4 4 3 2 2 1 3 5

Product Routing [2]: Shipping Truck:
2 2 1 3 1 1 3 3 4 4 5 4 4 5 5 5 5

Product Routing [3]: Number of Total Products transferred:
68 132 65 12 252 33 81 104 98 189 9 38 162 138 128 83 128

TEST SET 20

Receiving Truck Sequence:
1 2 6 1 5 2 3 1 2 4 5 2 1 6 3

Shipping Truck Sequence:
1 5 4 3 6 2

Product Routing [1]: Receiving Truck:
1 2 6 6 1 1 5 5 2 3 3 1 2 4 4 5 2 1 6 3

Product Routing [2]: Shipping Truck:
1 5 4 6 4 3 4 3 3 3 6 6 6 6 2 2 2 2 2 2

Product Routing [3]: Number of Total Products transferred:
204 180 191 6 5 10 128 12 91 76 240 106 45 60 220 220 74 45 83 24

REFERENCES

Baker, R. Kenneth "Introduction to Sequencing and Scheduling", *John Wiley & Sons*, New York, 1974.

Cooke, James Aaron "Do you have what it takes to Cross Dock?", *Logistics Management, September*, 47-50, 1996.

Forger, Gary, "UPS starts World's Premiere Cross-Docking Operation", *Modern Materials Handling, November*, 36-38, 1995.

Glover, Fred and Laguna, Manuel "Tabu Search", *Kluwer Academic Publishers*, Boston, 1997.

Morton, E. Thomas and Pentico, W. David "Heuristic Scheduling Systems with Applications to Production Systems and Project Management", *John Wiley & Sons, Inc.*, New York, 1993.

Rohrer, Matthew "Simulation and Cross Docking", *Proceedings of the 1995 Winter Simulation Conference"*, 846-849, 1995.

Schaffer, Burt "Cross Docking can Increase Efficiency", *Automatic I.D. News, 14(8) July*, 34-37, 1998.

Schwind, Gene F " Considerations for Cross Docking", *Material Handling Engineering, 50(12) November*, 47-51, 1995.

Schwind, Gene F. "A Systems Approach to Docks and Cross Docking", *Material Handling Engineering, 51(2) February*, 59-62, 1996.

Witt, Clyde E. "Crossdocking: Concepts Demand Choice", *Material Handling Engineering, 53(7) July*, 44-49, 1998.

Wurz, Al "Cross Docking is Workable Today!", *Automatic I.D. News, 10(5) May*, 56-57, 1994.

ACKNOWLEDGMENTS

I would like to reserve this space to thank all people who made this study and work possible. First, and most importantly, I would like to thank my major professor and Dean of College of Engineering at Louisiana State University, Dr. Pius J. Egbelu, for all his advice, guidance, supervision, support and encouragement throughout my research work. I am very honored to become his student.

I would like to express my sincere appreciation to my committee, Dr. Timothy Van Voorhis, for his advice and help in the mathematical modeling part of my research. I would also like to express my sincerest gratitude to my committee, Dr. Douglas Gemmill, Dr. Sarah Ryan and Dr. Yoshinori Suzuki for their guidance and helpful comments. They truly made this study better. I would like to thank my undergraduate professor Yunsun Park in Korea who has encouraged me to study continuously.

I would like to thank my parents for all their help and support that allowed me to focus on my study. Without their support, none of this could have taken place. And finally, above all, I would like to thank my wife, Sejin, for her sacrifices, patience, support and love and for just being there when I needed her. I dedicate this work to my parents, wife and son, Seungsoo.

VITA

NAME OF AUTHOR: Wooyeon Yu

DATE AND PLACE OF BIRTH: January 21, 1969, Seoul, Korea (ROK)

EDUCATION:

May 2002	**Iowa State University**, Ames, IA **Ph.D** in Industrial and Manufacturing Systems Engineering *Dissertation*: "Operational strategies for cross docking systems" *Major Professor*: Dr. Pius J. Egbelu
December 1997	**Iowa State University**, Ames, IA **M.S.** in Industrial and Manufacturing Systems Engineering *Thesis*: "Design of a zone-based tandem layout for automated guided vehicle systems" *Major Professor*: Dr. Pius J. Egbelu
February 1992	**Myong Ji University**, Kyoung-Ki Do, Korea (ROK) **B.S.** in Industrial Engineering

TEACHING EXPERIENCE:

2001–2001	**Iowa State University**, Ames, IA **Teaching Assistant** for the "Introduction to Manufacturing Processes and Specifications" course.
2000–2001	**Iowa State University**, Ames, IA **Teaching Assistant** for the "Solidification Processes" course.
2000–2000	**Iowa State University**, Ames, IA **Instructor** for the "Engineering Economics" course.
2000–2000	**Iowa State University**, Ames, IA **Teaching Assistant** for the "Manufacturing Systems Engineering" course.
1997–1999	**Iowa State University**, Ames, IA **Teaching Assistant** for the "Engineering Economics" course.

PUBLICATIONS:

Wooyeon Yu and Pius J. Egbelu, "Design of a Variable Path Tandem Layout for Automated Guided Vehicle Systems," to appear in *Journal of Manufacturing Systems*.

AWARD:

2002	**Finalist**: 2nd Tefen/IIE Graduate Student Competition for Excellence in Industrial Engineering. (in progress)