Analysis and Improvement of Cross-dock Operations in Less-than-Truckload Freight Transportation Industry

Xiangshang Tong

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Industrial and Systems Engineering

Kimberly P. Ellis, Chair
Ebru K. Bish
C. Patrick Koelling
Deborah F. Cook

August 11, 2009
Blacksburg, Virginia

Keyword: Cross-dock Operations, Less-than-Truckload, Trailer-to-door Assignment Problem, Freight Sequencing Problem, Simulation

Analysis and Improvement of Cross-dock Operations in Less-than-Truckload Freight Transportation Industry

Xiangshang Tong

(ABSTRACT)

The less-than-truckload (LTL) transportation industry is highly competitive with low profit margins. Carriers in this industry strive to reduce costs and improve customer service to remain profitable. LTL carriers rely on a network of hubs and service centers to transfer freight. A hub is typically a cross docking terminal in which shipments from inbound trailers are unloaded and reassigned and consolidated onto outbound trailers going to the correct destinations. Freight handling in a hub is labor intensive, and workers must quickly transfer freight during a short time period to support customer service levels. Reducing shipment transfer time in hubs offers the opportunity to reduce labor costs, improve customer service, and increase competitive advantages for carriers.

This research focuses on improving the efficiency of hub operations in order to decrease the handling costs and increase service levels for LTL carriers. Specifically, the following two decision problems are investigated: (1) assigning trailers to dock doors to minimize the total time required to transfer shipments from inbound trailers to destination trailers and (2) sequencing unloading and loading of freight to minimize the time required by dock employees.
The trailer-to-door assignment problem is modeled as a Quadratic Assignment Problem (QAP). Both semi-permanent and dynamic layouts for the trailer-to-door assignment problem are evaluated. Improvement based heuristics, including pair-wise exchange, simulated annealing, and genetic algorithms, are explored to solve the trailer-to-door assignment problem. The freight sequencing problem is modeled as a Rural Postman Problem (RPP). A Balance and Connect Algorithm (BCA) and an Assign First and Route Second Algorithm (AFRSA) are investigated

and compared to Balanced Trailer-at-a-Time (BTAAT), Balanced Trailer-at-a-Time with Offloading (BTAATWO), and Nearest Neighbor (NN).

The heuristics are evaluated using data from two LTL carriers. For these data sets, both the total travel distance and the transfer time of hub operations are reduced using a dynamic layout with efficient freight sequencing approaches, such as the Balance and Connect Algorithm (BCA), the Balanced Trailer-at-a-Time with Offloading (BTAATWO), and the Nearest Neighbor (NN). Specifically, with a dynamic layout, the BCA reduces travel distance by 10% to 27% over BTAAT and reduces the transfer time by 17% to 68% over BTAAT.

A simulation study is also conducted for hub operations in a dynamic and stochastic environment. The solutions from the door assignment and freight sequencing approaches are evaluated in a simulation model to determine their effectiveness in this environment. The simulation results further demonstrate that the performance measures of hub operations are improved using a dynamic layout and efficient freight sequencing approaches.

The main contributions of this research are the integer programming models developed for the freight sequencing problem (FSP), based on the Rural Postman Problem (RPP). This is the first known application of the RPP for the FSP. Efficient heuristics are developed for the FSP for a single worker and for multiple workers. These heuristics are analyzed and compared to previous research using industry data.

Acknowledgements

First and foremost, I would like to thank my advisor Dr. Kimberly Ellis. I was extremely fortunate to work with Dr. Kimberly Ellis, and I have learned so much from her research methodology as well as technical expertise. Without her guidance, inspiration, patience, and generous support through my years at Virginia Tech, my Ph.D. endeavor was just a dream.

I also want to thank my committee members: Dr. Ebru Bish, Dr. P. Koelling, and Dr. D. Cook for their willingness to serve on my committee for their encouragement and supportive feedback on my research at Virginia Tech.

I want to thank Lovedia Cole, Hannah Swiger, Kim Ooms, and many other faculty members and staff in the ISE department of Virginia Tech for their excellent assistance in helping me to finish this long process. I also want to thank many friends at ISE (Liming, Lixin, Chenbin, Weipin, Jin, Juqi, Lingrui, Susan, Kihuan …) for their friendship to make this journey bearable.

Finally, I would like to thank my wife Hailan, my parents and my wife's parents. Without their love and support, I would not have come this far.

Table of Contents

List of Tables

List of Figures

Chapter I

Introduction

Freight transportation plays a key role in today's national and global economy. Freight transportation allows production and consumption to take place at geographically dispersed locations that are several hundreds or thousands of miles away from each other. In 2008, logistics costs in the United State increased to $1.18 trillion (or 9.5% of nominal GDP), with 62% related to freight transportation costs (Wilson, 2008). In the freight transportation industry, cross-docking is increasingly used to reduce costs, decrease inventory time in the supply chain, and improve customer service.

With cross-docking, freight is transferred from inbound trailers or rail cars to outbound trailers or rail cars with no intermediate storage (or minimal storage). Cross-docking is used in hub-and-spoke networks where freight is transported to a central hub and then sorted for delivery to multiple destinations (Napolitano, 2008). Cross-docking is also used in consolidation arrangements, where a variety of smaller shipments are combined into a larger shipment for more efficient transportation, and in deconsolidation arrangements, where large shipments are separated into smaller shipments for ease of delivery (Napolitano, 2008).

In 2008, Saddle Creek Corporation commissioned an independent survey of logistics professionals in warehousing, distribution and transportations. Of 547 respondents, 52% currently cross-dock freight and 13% plan to begin cross-dock operation in the next 18 to 24 months (Napolitano, 2008). Minimal research has been conducted, however, on improving cross-dock operations. Less-than-truckload (LTL) carriers serve the market between parcel carriers and full truckload (FTL) carriers. LTL carriers have identified an opportunity to improve cross-docking for hub operations and increase customer service levels. This chapter introduces the LTL industry, describes cross-docking operations at LTL hubs, provides motivation for the research, and describes the organization of this dissertation.

1.1 LTL Industry Operations

LTL freight is often routed through a network of service centers and hubs as shown in Figure 1.1. Local pickup and delivery drivers usually have set routes which they travel every day or several times a week during the day. When drivers fill their trailer or complete their assigned routes, they proceed to the service terminal to deliver the freight. The trailer is unloaded and the individual shipments are then weighed and inspected to verify their conformity to the description contained in the accompanying paperwork. Next, the freight is loaded onto an outbound trailer which transports the freight to a hub or to the delivering terminal, usually during early evening. At the hub, freight from different service centers is consolidated onto common trailers to better utilize the truck resource during the night. The outbound trailers then transport the freight to the appropriate service center. When the freight arrives at the service center and is unloaded (usually in the morning), a local pickup and delivery driver begins the delivery routes. An LTL shipment may be handled multiple times before final delivery is accomplished.

Transit times for LTL freight are longer than for FTL freight. The main advantage of using an LTL carrier is that a shipment may be transported for a fraction of the cost of hiring an entire truck and trailer for an exclusive shipment. Also, a number of accessorial services are available from LTL carriers, which are not typically offered by FTL carriers.

The marginal cost of a package in a shipment is defined as the incremental increase in cost by adding a package to that shipment. Since the marginal cost of a package in a shipment on a trailer with excess capacity is very small, any policy that increases package densities reduces average cost per package per trip. With the service centers (spokes) feeding the packages to the hubs, the hub-and-spoke network configuration increases package densities on the inter-hub routes. Therefore, hub-and-spoke network operations are widely used in LTL industry.

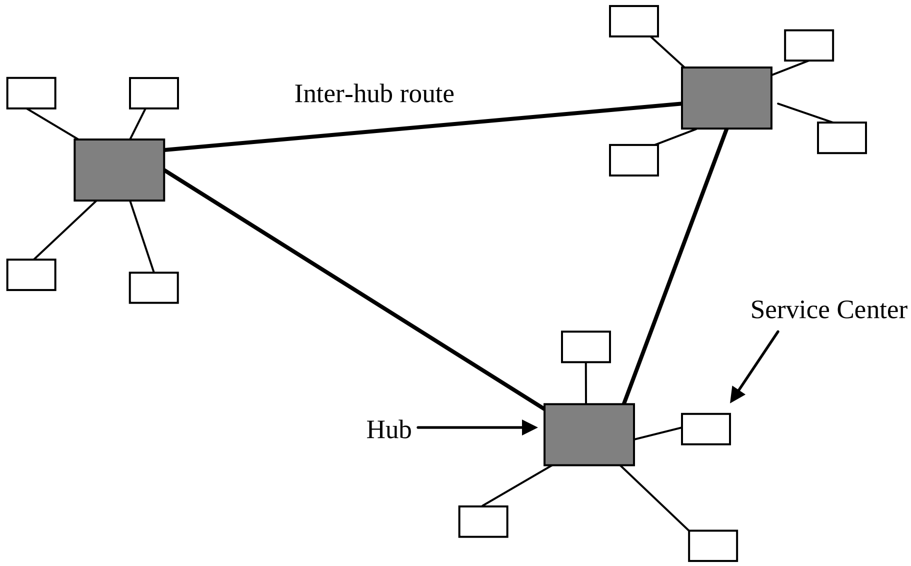


Figure 1.1 Hub-and-Spoke Networks

1.2 LTL Hub Layout and Operations

At an LTL hub, shipments are transferred from incoming to outgoing trailers. The shipments may also be classified, consolidated, and possibly stored for a short time. For LTL terminals, material handling is typically labor intensive (e.g. loads are moved by employees on forklifts). Packages often have different sizes, and automated material handling is difficult to incorporate.

An LTL hub has multiple doors (or gates) which are used for loading and unloading. Receiving doors, often designated as strip doors, are used for unloading the inbound freights. Shipping doors, often designated as stack doors, are used for loading the outbound freights. Four common shapes of the LTL hubs include I, H, L, and T terminals as illustrated in Figure 1.2.

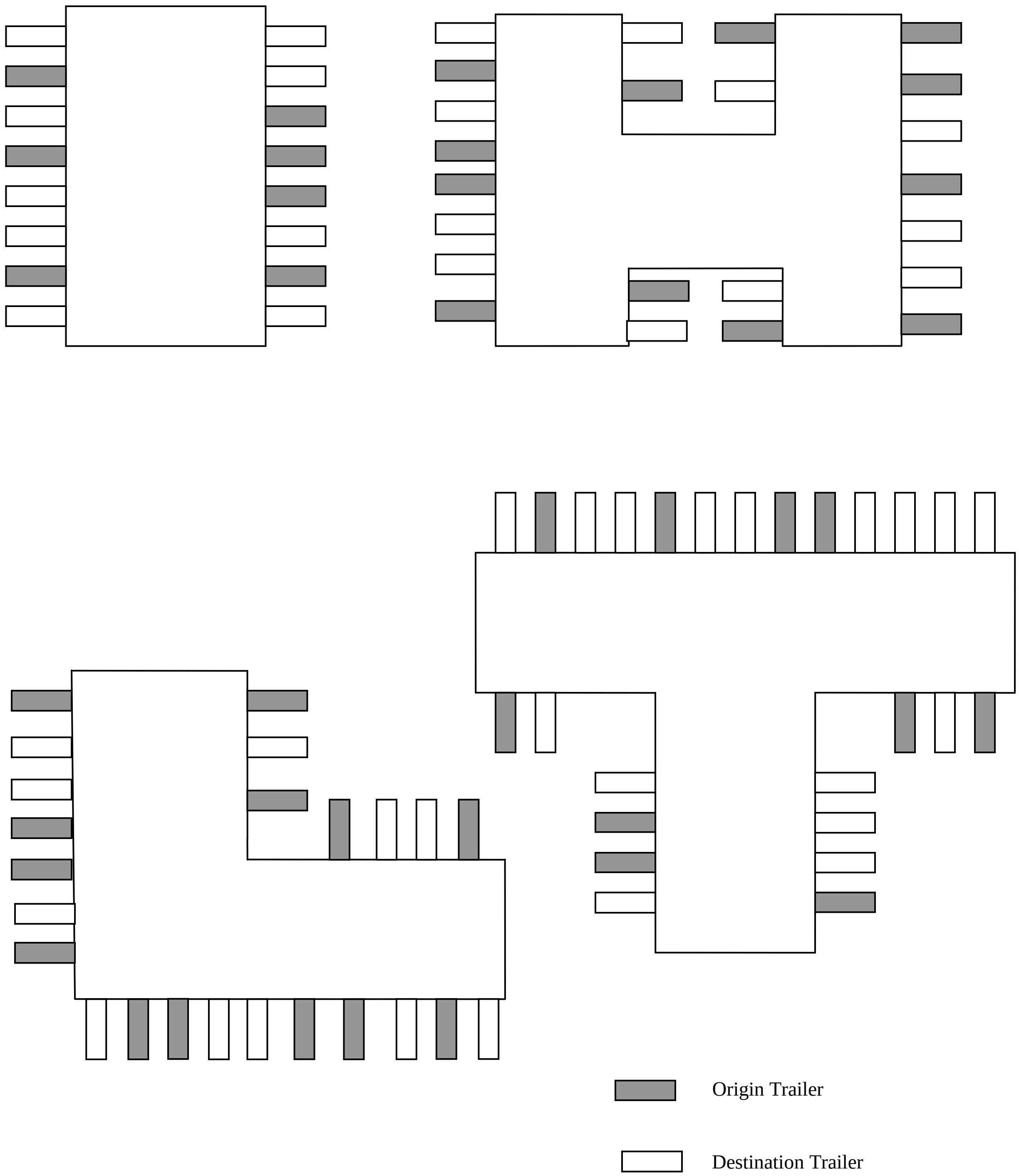


Figure 1.2 Common Shapes for LTL Hubs

At a hub, a dock supervisor decides how origin trailers and destination trailers are assigned to the docks. The dock supervisor also decides which trailers or shipments are assigned to which dock worker and the sequence the dock worker will unload the shipments from origin trailers and load them on destination trailers. Each shipment may consist of multiple handling units, where each handling unit requires one trip by a dock worker. These decisions are described in the following sections.

1.2.1 Assignment of Trailers to Dock Doors

Currently, for many LTL hubs, the assignment of trailers to dock doors is based on historical data and extensive experience, rather than actual shipping operations occurring that day. For example, the destinations trailers may be assigned to dock doors based on geography (such that Roanoke and Richmond doors are adjacent). Alternatively, the average number of shipments to a specific destination during a period (such as 6 months) may be collected and used for assigning destination trailers to the dock doors. The destination trailers are relatively fixed during a period. The origin trailers are assigned to the doors as they arrive to the hub based previous experience or on available doors. To increase hub efficiency, the assignment of trailers to doors could be based on the actual shipments for a given day, rather than using historical shipment data. In this research, alternative approaches for assigning trailers to dock doors are explored.

1.2.2 Unloading and Loading the Shipments

After the trailers (origins and destinations) are assigned to the dock doors, typically a dock worker is assigned to a trailer to remove a shipment and transport it on a forklift to the appropriate destinations trailer. The dock worker then returns to the origin trailer with an empty forklift to unload the next shipment on the manifest. This unloading process, referred to as trailer-at-a-time, continues until an entire trailer is unloaded. The trailer-at-a-time approach is relatively straightforward to plan and execute but often results in excess movements when the worker is traveling with an empty forklift. Alternative approaches for trailer-at-a-time may reduce the distance that the worker travels with an empty forklift. For example, the worker may unload a shipment from an origin trailer, transport it on a forklift to the destination trailer, and then unload the shipment from another nearby origin trailer rather than return to the first origin trailer. This research will develop an efficient algorithm to optimize the sequence of unloading and loading freights to increase efficiency for dock worker.

1.3 Challenges in LTL Industry

In the competitive freight transportation industry, LTL carriers focus on improving profit margins (Kent, 2007). LTL carriers also strive to reduce the total cost associated with freight transportation, handling, and storage. The service level in LTL industry greatly influences customer satisfaction and has a major impact on revenues. The most commonly used service level measurement in LTL industry is the delivery time, normally defined as the elapsed time between when the freight is received from the shipper and the time the freight is delivered to the consignee. This research strives to reduce handling time and costs and improve service levels for hub operations.

1.4 Research Motivation and Objectives

Handling freight in an LTL hub is labor intensive and therefore expensive since workers must quickly unload and load a variety of freight. Many LTL carriers operate hundreds of hubs, and improvements in operations at one hub offer opportunities for improvements at all hubs. Therefore even small improvement in hub operations offers significant opportunities for the LTL carrier. This research is focused on improving the hub operations of LTL carriers and decreasing the handling costs of the hub operations. Specifically, the following two problems in the hub operations are investigated:

- assigning trailers to dock doors
- sequencing the unloading and loading of freight

Assigning trailers to dock doors involves optimizing the assignment of trailers to dock doors to minimize the total time required to load and unload the shipments from the trailers. This problem is referred to as the trailer-to-door assignment problem. Sequencing the unloading and loading of freight involves developing a sequence of unloading and loading operations for a dock employee to minimize the amount of time that employees are traveling. This problem is referred to as the freight sequencing problem. If multiple dock workers are transferring freight, then a related problem is assigning freight to workers in addition to sequencing the unloading and loading operations. This problem is referred to as the workload assignment problem. This research develops and compares solution approaches for the trailer-to-door assignment problem and the freight sequencing problem in a realistic environment.

1.5 Organization of the Dissertation

In this dissertation, the literature for LTL hub operation problems and solution approaches is summarized in Chapter 2. Chapter 3 provides the framework of the research methods. Chapter 4 presents mathematical models for both the trailer-to-door assignment problem and freight unloading and loading sequencing problem. Chapter 5 discusses solution approaches to the trailer-to-door assignment problem. Chapter 6 focuses on solution approaches to the freight unloading and loading sequencing problem. Chapter 7 develops a simulation model to implement trailer-to-door assignment and freight sequencing approaches in a more realistic hub operations environment. Chapter 8 summaries the results of this research and feature research directions.

Chapter II

Literature Review

The literature for the trailer-to-door assignment problem and the freight sequencing problem of the LTL hub operations is reviewed in this chapter. The literature is reviewed in three parts: (1) hub operations, (2) Quadratic Assignment Problem and algorithms, and (3) Rural Postman Problem and algorithms.

2.1 Hub Operations

Some of the earliest work on the trailer-to-door assignment problem for cross-dock operations was published by Peck (1983). Peck (1983) described and simulated a "full floating dock", in which terminal space is allocated on a continuous basis. Previously, the layout of an LTL terminal was fixed, and the dock was partitioned into sections by geographical area. Peck, however, proposed a "floating dock" which is changed frequently. Peck (1983) presented a Greedy Balance Algorithm (GBA) to assign doors to minimize freight transfer time. The GBA first finds a feasible solution and then uses a pair-wise exchange approach to improve the solution. Peck also developed a detailed simulation of activities at an LTL hub. The simulation model captured the physical location of doors of the hubs, the characteristics of the material handling system, a description of the freight, and the activities of workers.

The trailer-to-door assignment problem was first formulated as an integer programming (IP) problem by Tsui and Chang (1990). The objective of the problem is to find the optimal arrangement of inbound and outbound doors and the most efficient assignment of destinations to outbound doors, such that the weighted distances between incoming and outgoing trailers are minimized. An initial local search procedure was used to find a local optimal solution. This initial local search approach is very sensitive to starting solutions. Tsui and Chang (1990) later developed an optimal algorithm using branch-and-bound techniques. The algorithm was tested on randomly generated data, and the results showed that computation time increases dramatically as the size of the problem increases.

Gue (1995) developed a model for the trailer-to-door assignment problem in LTL cross-dock operations to minimize travel costs, congestion costs, and interference costs. The solution approach combines traditional swap heuristics and queuing analysis. Gue (1995) stressed that minimizing the weighted distance alone might create congestion and interference of material handling equipment in LTL hub. Gue (1995) provided freight flow models such as average trailer and specific trailer to test the quality of a solution. In addition, Gue (1995) addressed the congestion issue in a hub by proposing the pressure concept, analogous with a force, on a dock door. Pressure is the freight flow bound for a destination, per unit area in front of that destination, plus freight flow from the adjacent zones when the adjacent zones are above their maximum allowable pressure. Using the maximum pressure constraint Gue (1995) determined when a destination will be assigned two adjacent doors. Gue (1995) developed Two-Block Algorithm which makes an initial assignment of trailers to dock doors based on travel distance, and then used pair-wise exchange to determine improved assignments.

Since the underlying structure of the trailer-to-door assignment problem is the NP-hard QAP, various heuristic approaches have been used to solve the trailer-to-door assignment problem. Bermudez and Cole (2002) presented a genetic algorithm (GA) to assign trailers to doors in the hub operations to minimize total travel distance. Several different GA parameters were tested in various combinations to determine which parameters are most appropriate for three different sized hubs. Although definitive rules for which GA parameters to use are not provided, several suggestions are provided based on observation of computational results. The results from the GA approach were compared to results obtained using pair-wise exchange and the GA solutions outperformed the pair-wise exchange solutions.

Bozer and Carlo (2008) presented a linear mixed integer programming formulation to minimize total distance for the trailer-to-door assignment problem for an LTL cross-dock. A simulated annealing (SA) heuristic was used to solve a general rectilinear-quadratic problem for a semi-permanent layout where doors are fixed over a planning horizon. Using the data sets from industry, the authors indicate the door assignment obtained from SA heuristic outperforms industry practice.

Brown (2003) modeled the trailer-to-door assignment problem as a QAP and solved the problem using pair-wise exchange. Brown (2003) also classified solutions for the trailer-to-door assignment problem as semi-permanent layouts or dynamic layouts. With a semi-permanent layout, the destination trailers are assigned to doors for three to six month (usually based on historical data) and the origin trailers are assigned to specific dock doors on a nightly basis depending on freight flow. With a dynamic layout, both the origin and destination trailers are assigned to dock doors on a nightly basis depending on freight flow.

The airport layout problem in the airline industry is similar to the trailer-to-door assignment problem. In airport hubs, passengers walk between connecting flights depending on gate assignment in the airport terminal. Mangoubi and Mathaisel (1995) evaluated two approaches for gate assignment, with the objective of minimizing walking distance for passengers. Bihr (1990) also develops a solution for gate assignments in airport terminals to minimize walking distance for passengers.

Most of the LTL hub operations research reviewed assumes the trailer-at-a-time approach for the freight sequencing problem with the exception of Brown (Brown, 2003). This research extends Brown's research to explore alternative approaches for the freight sequencing problem.

The freight unloading and loading sequence problem is very similar to the component placement sequence problem in printed circuit board (PCB) assembly industry. The component placement sequence problem can be described as follows: many components from the edge of the board are picked by a numerically controlled automated machine and placed on the board in pre-determined positions. The problem is to develop a sequence of picking and placing the components that minimize the total operation time to achieve the goal of maximum throughput of the assembly line. While this problem is widely modeled as a Traveling Salesman Problem (TSP) in the literature, Ball and Magazine (1988) modeled this problem as a Rural Postman Problem (RPP). Machine head movements can be partitioned into required and non-required movements, which correspond to the required and non-required arcs in the RPP network. The goal is to find the Euler tour that minimizes the total distance. Ball and Magazine (1988) developed an efficient heuristic and compared to a derived lower bound.

This component placing sequence problem is similar to the freight unloading and loading sequence problem but it is different in the following way: The PCB numerically controlled machine has one arm and only one component can be placed at any given time. In the LTL hub, however, there are often multiple workers moving freight at the same time. The unloading and loading time and travel speeds in the LTL hub also have much higher variability and are harder to predict than in PCB assembly.

2.2 Quadratic Assignment Problem (QAP) and Applications

The underlying structure of the trailer-to-door assignment problem is a Quadratic Assignment Problem (QAP). The QAP was introduced by Koopmans and Beckman (1957) as a mathematical model for the locations of divisible economical activities. The QAP is often used to describe a location problem when n facilities are assigned to n locations with a cost proportional to the flow between the facilities multiplied with their distances. The objective is to allocate each facility to a location such that the total cost is minimized.

Many real life applications can be modeled as QAPs. A natural application in location theory was used by Dickey and Hopkins (1972) in a campus planning model. The problem was to determine the sites of n buildings on a campus, where b_{kl} is the distance from site k to l and a_{ij} is the traffic intensity between buildings i and j. The objective was to minimize the total weekly walking distance between the buildings.

In addition to facility location problems, the QAP structure appears in applications such as facility layout problems, backboard wiring, computer manufacturing, scheduling, process communications and turbine balancing. Burkard and Offermann (1997) show that the typewriter keyboard design can be modeled as a QAP. The problem is to arrange the keys on a keyboard to minimize the time needed to type some text.

Quadratic Assignment Problems remain among the hardest combinatorial optimization problems to solve as reflected by their computational complexity. Sahni and Gonzalez (1976) showed that the QAP is NP-hard. They also showed that finding an approximate solution within some constant factor from the optimum value cannot be done in polynomial time.

Since QAPs are NP-hard, only implicit enumeration methods are known for solving them optimally. Most of these approaches are branch-and-bound methods based on various bounding techniques and exclusion tests and cutting plane methods (Bazaraa and Sherali, 1980). Even today, however, it is not possible to solve QAPs to optimality for $n > 20$. Thus, heuristic methods are often employed. Heuristic for QAPs can be classified as construction methods and improvement methods.

Construction methods generate a feasible solution step by step (i.e. by assigning one index after the other according to certain criteria) until a permutation of the whole index set is obtained. For example, the heuristics described by Gilmore (1962) are construction methods. Usually, construction methods are easy to implement and have minimal computational times, but the quality of their solutions leaves much to be desired.

Improvement methods start with a feasible solution and try to improve it by exchanging single assignments. In the pair-wise exchange algorithm, pairs of indices are chosen and images are interchanged. Among these improvement methods, a method developed by Heider (1972) is shown to be rather efficient. Burkard and Rendl (1983) applied a simulated annealing approach to solve QAPs and performed computational analysis of this method. Following this implementation, other simulated annealing approaches were proposed. Currently the approach proposed by Conolly (1990) appears to be the best performing in the computational time. Thonemann and Bolte (1994) proposed an improved SA algorithm for the QAP. The best known tabu search algorithm for QAP is the robust tabu search (RTS) algorithm of Taillard (1991). This algorithm is based on the 2-opt best improvement local search algorithm. Genetic algorithms (GAs) are also used to solve the QAPs. Kochar et al. (1998) apply GAs to the facility layout problem (a version of the QAP). They find solutions within 1% to 5% of the best-known solutions. Nissen (1994) presents an evolutionary strategy (not strictly a GA) for solving quadratic assignment problems. He found that approaches modeled after biological natural selection often outperform traditional heuristics such as 2-opt.

2.3 Rural Postman Problem (RPP) and Applications

The underlying structure of the freight sequencing problem may be viewed as a Rural Postman Problem (RPP). The RPP is related to the Chinese Postman Problem (CPP) in the literature. The CPP is described by Guan (1962) as follows: "A mail man has to cover his assigned segment before returning to the post office. The problem is to find the shortest walking distance for the mailman." In contrast to another well known routing problem, the Traveling Salesman Problem (TSP), the requirement in the CPP is to cover all the arcs in the graph instead of covering all the vertexes (i.e., cities) in TSP.

The RPP is the general case of the CPP in the sense that only some arcs need to be traveled. Strictly, the CPP is defined on a graph $G = (N, A)$, where N is the vertex set, A is the arc set, and a nonnegative cost matrix $C = (c_{ij})$ is associated with A. The arcs can be directed, undirected, or both. CPP seeks a minimum cost closed walk on all arcs of A. The RPP is required to traverse only a subset $R \subseteq A$ of arcs. When $R = A$, that is the CPP. There are fewer practical contexts where it is necessary to service all arcs of a network. Hence, many arc routing applications are modeled as the RPP. The freight sequencing problem belongs to the RPP since the required arcs are connected by many non-required arcs. The dock worker must travel all the required arcs (exactly once) but not all the non-required arcs.

Many applications of the RPP are found in the literature where streets or roads have to be traversed for maintenance, garbage collection, milk or mail delivery, school bus transportation, parking meter collection, electric meter reading, electrical lines and gas mains inspection. Bodin and Kursh (1978) describe a computer assisted system for the routing and scheduling of street sweepers, together with computational experiences derived from two pilot studies in New York City and Washington, D.C. In addition to the algorithm, the authors describe important preprocessing steps, such as data preparation and network verification. An important aspect of street sweeping is that particular streets can only be swept at certain times because of parking regulations. Thus, the problem involves selecting which streets to include in each route so that workloads are balanced and streets are serviced at suitable frequencies.

Haslam and Wright (1991) describe an algorithm used for the design of highway snow and ice control in Indiana. Since Indiana roads are classified in several categories with varying priority levels for snow clearance, the problem can be modeled as a hierarchical RPP.

Striker (1970) developed a computerized arc routing algorithm for the urban waste collection problem which is also a RPP application. Many postman delivery problems are modeled as an RPP. In these problems, the arcs are partitioned into clusters and depots are located to use as starting points for the postmen's route. Clusters must be balanced and correspond to the maximum mail volume that may be carried at the same time.

School bus routing is another major application area of the RPP (Bennett & Gazis, 1972). In addition to the usual traffic restrictions, several constraints must be satisfied, including time windows, minimum and maximum number of passengers in the bus at any given time, student mix, and maximum time spent by any student on the bus.

Polynomial algorithms exist for the CPP but the RPP is NP-hard (Lenstra & Kan, 1976). Thus, heuristic solution approaches are generally employed. Frederickson (1979) has solved the undirected RPP with a heuristic that works along the lines of the Christofides heuristic (Christofides, 1976) for the undirected traveling salesman problem. The heuristic has a worst case ratio of 3/2 of the optimal solution.

Christofides et al. (1986) proposed a heuristic for the directed RPP. The basic idea for this heuristic is to solve the transportation problem first and connect the resulting disconnected graph using shortest spanning tree algorithm to obtain an augmented graph. Finally, a Euler tour algorithm is applied to find the Euler tour on the augmented graph. This heuristic was also studied by Ball and Magazine (1988).

Hertz et al. (1999) proposed a local search heuristic for the undirected RPP based on the observation that if a tour contains a chain *C* of non-required edges, then it can eventually be reduced by replacing *C* with a shortest chain linking the endpoints of *C*. This heuristic can be also adapted and extended for the directed and mixed RPP.

The Rural Postman Problem with multiple postmen is often referred to as a *k* Rural Postman Problem (*k* RPP). Frederickson et al. (1978) formulated *k* person routing problems as min-max problems and developed a tour splitting algorithm which obtained approximate solutions for multi-routing problems from good approximate solutions of simple routing problems.

Another family of routing problems similar to the *k* RPP is the Capacitated Arc Routing Problem (CARP) or Capacitated Vehicle Routing Problem (CVRP). A fleet of *m* homogeneous capacitated vehicles are based at a depot and the problem is to determine a minimum cost traversal of all arcs by the vehicles subject to a capacity constraint for the vehicles. Christofides (1973) proposed a simple constructive method for the CARP. The heuristic gradually constructs feasible cycles and removes them from the graph without disconnecting the graph. When feasible cycles can no longer be found, an Euler tour is constructed on the graph.

The Augment-Insert Algorithm was suggested by Pearn (1991) for the CARP. This algorithm is to first gradually insert all arcs as long as possible into feasible cycles connected to the depot by cost or demand criterion and then include all remaining arcs in the existing cycles by savings criterion.

The Cluster-First-Route-Second heuristic by Benavent et al. (1990) partitions the arcs into clusters and then solves the uncapacitated RPP on each cluster. On the other hand, Route-First-Cluster-Second heuristic by Ulusoy (1985) first constructs a large Euler tour cover all edges and then partitions the Euler tour into feasible vehicle routs.

In recent years, several meta-heuristics are proposed for the CARP. Eglese (1994) has developed a simulated annealing approach for the CARP with multiple depots and several constraints. A solution of the CARP is defined as a set of trees rooted at the depot node. Neighborhood solutions are obtained by performing simple modifications on at most two trees. Hertz et al. (2000) have also developed a tabu search algorithm for the CARP.

2.4 Summary of Literature Review

The underlying structure of the trailer-to-door assignment problem is related to the QAP, and the underlying structure of the freight sequencing problem is similar to the RPP. This chapter reviewed the literature for LTL hub operations, as well as the literature for the QAP and RPP. Most research on LTL hub operations focuses on the trailer-to-door assignment problem rather than the freight sequencing problem (with exception of Brown, 2003). The trailer-to-door assignment problem is modeled as the QAP. Since the QAP is NP-hard, various heuristic approaches are used to solve the problem. Construction algorithms are often used to develop an initial solution and an improvement algorithm (i.e., pair-wise exchange) is often used to improve the initial results. Meta-heuristics have recently been used to solve the QAP in the literature, although the results of the meta-heuristics are highly problem dependent. This research will implement a simulated annealing algorithm and a genetic algorithm to address the trailer-to-door assignment problem.

All of the research on the LTL hub operations (except Brown, 2003) assumes that a dock worker unloads a trailer until that trailer is empty. Brown (2003) identifies a new problem, freight unloading and loading sequencing problem, in the LTL hub operations by assuming that a dock worker can start unloading a trailer before fully completing another trailer. Brown (Brown, 2003) employed several heuristics to find the improved freight unloading and loading sequences for a dock worker. This research will extend Brown's research to model the freight unloading and loading sequence problem as an RPP and explore heuristics to find a better solution.

Most literature on the LTL hub operations models the trailer-to-door assignment problem as a QAP without considering the dynamic and stochastic characteristics in the hub operations. Gue (1999) considered the dynamic arrivals of the incoming trailers as well as the congestion in the hub operations. Brown (2003) used simulation model to evaluate the results from the heuristic approaches. This research will also use simulation models to evaluate the results from the optimization models.

Chapter III

Research Methodology

This research focuses improving the efficiency of cross-dock operations for LTL hubs. Two operational decisions made on a daily basis by hub supervisors include: assigning trailers to dock doors and sequencing the unloading and loading of freight. The underlying structure of the trailer-to-door assignment problem is a Quadratic Assignment Problem (QAP) and the underlying structure of the freight sequencing problem is a Rural Postman Problem (RPP). Since both the QAP and the RPP are NP-hard, various heuristic methods are implemented in this research to develop solutions for the two problems.

The solution to the trailer-to-door assignment problem affects the solution to the freight sequencing problem. Also, the approach used for freight sequencing influences the preferred assignment of trailers to doors. Thus, ideally the problems could be addressed simultaneously. Due to their complexity, however, the problems are addressed independently. The trailer-to-door assignment problem is solved and the solution is used by the freight sequencing problem, which is then solved. Finally, both the solutions are analyzed using a simulation model. The research methodology framework is illustrated in Figure 3.1.

3.1 Assignment of Trailers to Dock Doors

The approaches for assigning trailers to dock doors can be classified as fixed assignments, *semi-permanent* assignments, and *dynamic* assignments. In the first category, trailers are generally assigned to the same doors on a regular basis. For example, trailers destined for Roanoke, Richmond, or Charlotte may always assign to the same door. The trailers may even be assigned on geographic proximity such that Richmond and Roanoke are assigned to adjacent doors. Although this approach is easy to implement at the hub, the shipping patterns change over time. If the assignment of doors is adjusted accordingly, the shipments can be processed with fewer resources. On the other hand, flexible assignment approaches assign the trailers to dock doors according to the information of the shipments.

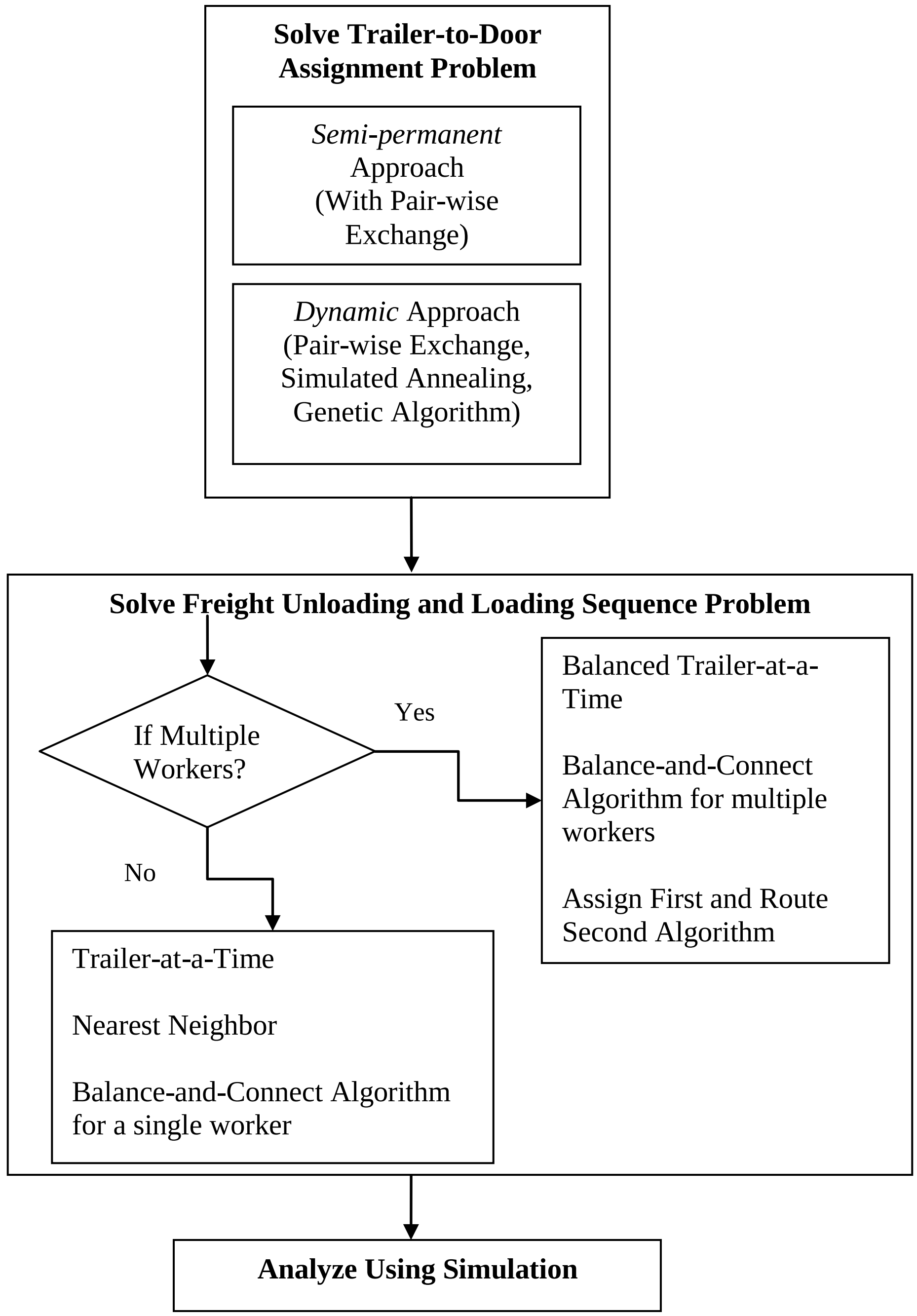


Figure 3.1 Research Framework

If flow of shipments between an origin and destination is relatively high, it is beneficial in terms of reducing traveling distance of workers if these two trailers are assigned to the close dock doors. There are two approaches in the flexible assignment approach category: *semi-permanent* layout and *dynamic* layout. The *semi-permanent* layout approach, uses the historical shipments data to assign the trailers to dock doors. In this approach, the average numbers of shipments to a specific destination during a period (say, 6 months) are collected and used for assigning that destination to the dock door while the origin trailers are assigned to dock doors based on the daily characteristic of their shipments. After the period, the average numbers of shipments to the destinations are re-collected and the assignment of the destination trailers to the dock doors is revised. The *dynamic* layout approach uses daily shipment data to assign both origin and destination trailers to the dock doors. Using daily shipments data to assign trailers to dock doors offers the opportunity to improve efficiency compared to the *semi-permanent* layout assignment approach which relies on historical shipment data.

To address the trailer-to-door assignment, heuristic based improvement methods are explored for the *semi-permanent* layout and the *dynamic* layout approach. When addressing the trailer-to-door assignment problem, the assumption is that that the dock workers fully complete one origin trailer before completing another origin trailer (using trailer-at-a-time). The heuristics are analyzed using industry data from LTL carriers.

3.2 Unloading and Loading Shipments

After the assignment of trailers to dock doors is found, the positions of both origin and destination trailers are known and the sequence that dock workers unload and load the shipments needs to be determined. For the freight sequencing problem for a single worker, the Balance-and-Connect Algorithm is developed and compared to the trailer-at-a-time and the nearest neighbor approach (Brown 2003). For the freight sequencing problem for multiple workers, the Assign First and Route Second Algorithm and the Balance-and-Connect Algorithm are developed. For the Assign First and Route Second Algorithm, an intermediate assignment of workers to origin trailers is performed first. The general goal is to balance the workload assigned to the workers so that the make-span of completing the transfer of shipments is minimized. After the dock workers are assigned to origin trailers, the Balance-and-Connect

Algorithm for a single worker is used to determine the preferred sequences for unloading and loading shipments. For the Balance-and-Connect Algorithm for multiple workers, the assignment of workers to the origin trailers and the sequence of unloading and loading shipments for each worker are obtained at the same time. The freight sequencing problem for both single worker and multiple workers are analyzed using industry data from LTL carriers.

The solutions found in this stage together with the solutions from the trailer-to-door assignment, are analyzed using a simulation model to validate the effectiveness of those solutions in more realistic environment.

3.3 Simulation Study

In the simulation study, some assumptions from the previous heuristic approaches are relaxed (i.e., workers travel speed and the unloading and loading times are constant, no congestion in the hub, and all trailers are available at the beginning of the operations, etc). Dynamic and stochastic characteristics are incorporated into the hub operations study.

The following chapters describe the models and solution approaches for the trailer-to-door assignment problem, the models and solution approach for the freight sequencing problem, and the simulation model of cross-dock operations.

Chapter IV

Problem Descriptions and Mathematical Models

This research focuses on improving the efficiency of an LTL hub operation by minimizing the time to transfer shipments from origin trailers to destination trailers and reducing labor costs. One of the primary methods of reducing the shipment transfer time is to assign trailers to dock doors to minimize total worker travel distance by dock workers. Another way is to optimize the freight unloading and loading sequence for the dock workers. The formal problem statement and mathematical models for these two problems are presented in this chapter.

4.1 Trailer-to-Door Assignment Problem

An efficient assignment of trailers to dock doors at hubs can reduce the total time required to transfer shipments by placing inbound origin trailers close to associated destination trailers and reducing the travel distance required by the workers. Assume a set of n trailers and a set of n dock doors are available at an LTL hub. For each pair of doors, a travel distance is specified. For each pair of trailers, a flow is specified that captures the number of handling units of freight to transfer between two trailers. The objective is to assign each trailer to a dock door with the goal of minimizing the sum of the distances multiplied by the corresponding flows.

In this research, the following assumptions are used to address the trailer-to-door problem:

- Origin and destination trailers are available at the beginning and throughout the hub operations;
- Dock doors are available for all trailers;
- A trailer is assigned to only one dock door and a dock door will be assigned at most one trailer; and
- The travel speed for workers is constant, and the unload and load times for a handling unit are constant.

A formal mathematical model of the trailer-to-door assignment problem is developed using the following indices, parameters, and variables:

Indices:

i indices for origin trailers, i = 1, 2,…, n

j indices for destination trailers, j = 1, 2,…, n

k, l indices for dock doors, k, l = 1, 2,…, n

Parameters:

f_{ij} flow of freight between origin trailer i and destination trailer j

d_{kl} distance between door k and door l

Variables:

x_{ik} = 1 if origin trailer i is assigned to door k; 0 otherwise

x_{jl} = 1 if destination trailer j is assigned to door l; 0 otherwise

The problem can be formulated as an integer programming with a quadratic objective function.

$$\text{Minimize} \quad \sum_i \sum_j \sum_k \sum_l f_{ij} d_{kl} x_{ik} x_{jl} \tag{4.1}$$

Subject to:

$$\sum_k x_{ik} = 1 \qquad \forall\, i \tag{4.2}$$

$$\sum_i x_{ik} \le 1 \qquad \forall\, k \tag{4.3}$$

$$x_{ik},\, x_{jl} \in \{0,1\} \tag{4.4}$$

The objective function minimizes the total distance traveled to transfer shipments from each origin trailer i assigned to door k and destination trailer j assigned to door l. The constraints (4.2) and (4.3) ensure that each trailer is assigned to exactly one door and each door has at most one trailer assigned. The resulting formulation is a Quadratic Assignment Problem (QAP).

The quadratic form of the objective function can be linearized using the following linearization substitution techniques (Adams et al, 2007; Adams and Sherali 1990):

let $y_{ijkl} = x_{ik}\, x_{jl} \qquad \forall\ i<k,\ j\neq l.$

The resulting QAP is equivalent to a mixed-integer linear program with n^2 integer and $n^2(n\text{-}1)^2/2$ real variables and further $2n^2$ linear constraints (Adams, Guignard, Hahn & Hightower, 2007; Adams & Sherali, 1990):

$$\text{Minimize} \quad \sum_i \sum_j \sum_k \sum_l f_{ij} d_{kl} y_{ijkl} \tag{4.5}$$

Subject to:

$$\sum_k x_{ik} = 1 \qquad \forall\ I \tag{4.6}$$

$$\sum_i x_{ik} \le 1 \qquad \forall\ k \tag{4.7}$$

$$y_{ijkl} \le x_{ik} \qquad \forall\ i<k,\ j\neq l \tag{4.8}$$

$$y_{ijkl} \le x_{jl} \qquad \forall\ i<k,\ j\neq l \tag{4.9}$$

$$y_{ijkl} \ge x_{ik} + x_{jl} - 1 \qquad \forall\ i<k,\ j\neq l \tag{4.10}$$

$$x_{ik},\ x_{jl} \in \{0,1\},\ y \ge 0 \tag{4.11}$$

4.2 Freight Sequencing Problem

Given an assignment of trailers to dock doors and a single worker, the freight sequencing problem (FSP) determines an optimal sequence for a worker to unload shipments from origin trailers and load shipments on corresponding destination trailers to minimize the time to complete all shipments.

In the optimal sequence, unlike the trailer-at-a-time approach assumed by most researchers in the literature, the worker may not complete a trailer before starting to unload another origin trailer. The worker might unload a shipment from an origin trailer, transport it on a forklift to the destination trailer, and then unload the shipment from another trailer nearby.

The following assumptions are made in addressing the freight sequencing problem:

- Trailers are assigned to dock doors at the hub;

- The travel distance between dock doors is known;
- Each handling unit requires one trip;
- The speed of the forklift and unloading and loading times are constant.

As described in the literature review, the FSP can be modeled as a directed Rural Postman Problem (RPP). Given the pre-determined shipment flow data, the movements of forklifts from origin trailer doors to destination trailer doors are viewed as required movements since the transfer must occur. The movements of forklift from destination trailer to origin trailer are viewed as non-required movements since multiple paths are possible. In terms of a graph, the movements can be modeled as required arcs and non-required arcs.

Figure 4.1 illustrates a hub, where dark blocks represent the origin trailers and white blocks represent the destination trailers. The bold directed arcs represent forklift movements for transferring shipment handling units from origin trailer doors to destination trailer doors (which are required trips). The dash directed arcs represent the empty forklift travel from destination trailers to origin trailers (which are non-required trips that may be accomplished with alternate paths)

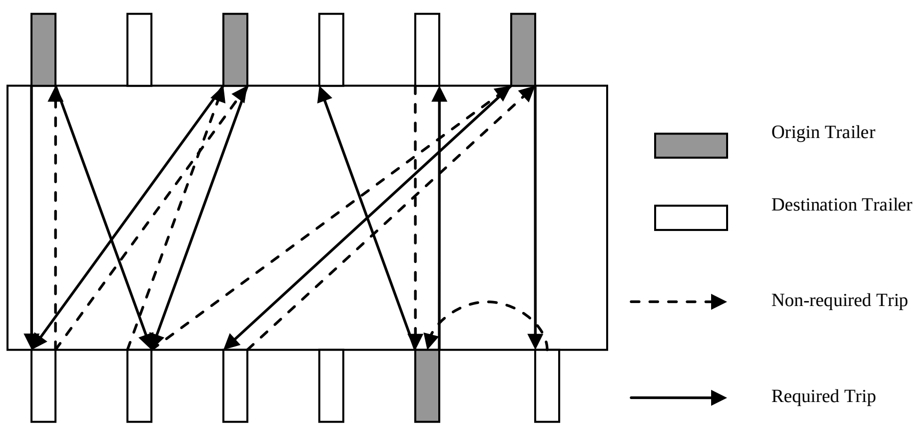


Figure 4.1 Required and Non-required Trips for Transferring Freight

Define a network $G = (N, A)$ where N is the set of all nodes representing the doors that are assigned to origin and destination trailers and A is the set of all directed arcs (n required paths plus the link between the ending points of all n paths). Let R be the set of all required directed arcs. Thus, R is subset of A. The arcs in R are required arcs and the other arcs in $A \backslash R$ are non-required arcs or deadhead arcs. The sub graph $G_R = (N, R \subseteq A)$ which includes the arcs of R is a required sub graph. The cost c_{ij} ($c_{ij} \geq 0$) of traversing each arc (i, j) is the distance between node i and node j.

The number of arcs linked to a node is called the degree of this node. Let node i be a door of the hub. On the sub graph $G_R = (N, R \subseteq A)$ the degree of node i is denoted as degree(i). If node i is occupied by a origin trailer, degree(i) is the number of arcs pointing out of node i, which corresponds to the number of shipments from node i to other nodes. If node i is occupied by a destination trailer, degree(i) is the number of arcs pointing into the node i, which corresponds to the number of required shipments into node i .

In the freight sequencing problem, one shipment may include multiple handling units, but a worker can only transfer one handling unit (such as a pallet) in each trip using a forklift truck. Sometimes carriers prefer that a single dock worker transfers all the handling units in a shipment. The number of required arcs in the directed RPP network could be either the number of shipments or the number of handling units. For example, assume a shipment from origin trailer 1 to destination trailer 2 has two handling units. If shipments are used for the required arcs, then the number of required arcs from node 1 to 2 is 1. If handling units are used for the required arcs, then the required arcs from node 1 to node 2 is 2. The example is illustrated in Figure 4.2.

The number of required arcs equals the number of shipment

The number of required arcs equals the number of handling unit

Figure 4.2 Value of d_i using Number of Shipments or Number of Handling Units

Notice that if the number of shipments is used to calculate d_i, the required arc only represents the last handling unit in that shipment. For the above example, arc (1,2) only represents the last handling units from 1 to 2. After the first handling unit is transferred from node 1 to node 2, the worker returns to node 1 for the second handling unit. So if the number of shipments is used to calculate d_i, the non-required arcs generated in the RPP model determine the return trips when a worker finishes the last handling unit in a shipment. On the other hand, if the number of handling units is used to calculate d_i, since the required arc represent each handling unit which needs to transferred, the non-required arcs generated in the RPP model determine the return trips for each handling unit. Using the number of handling units to calculate d_i provides more alternatives for return trips than using the number of shipments. Thus using handling units provides more opportunity for reducing travel distance in the RPP. In this research, handling units are used to calculate d_i for the required arcs.

A Euler cycle is a tour which starts at a node, traces each arc exactly once and ends at the starting node. A tour that ends at another node is called an open Euler trail. Graphs which allow the construction of Euler cycles or Euler trails are called Euler Graphs.

From graph theory, an Euler cycle on a graph is possible if and only if every vertex on the graph is of even degree (Raton, 2000). In the FSP, we assume that a worker starts at an origin trailer and ends at that origin trailer when all shipments are transferred. Thus, the problem is to find an Euler cycle for the worker where the number of arcs out of any node is equal to the number of arcs into that node.

4.3 Freight Sequencing Problem for a Single Worker

Using the RPP structure, a mathematical programming formulation is presented for the freight sequencing problem for a single worker. The following indices, parameters and variables are used in the model:

Indices and Sets:

i, j	indices of nodes (doors)
N	set of all nodes (doors)

Parameters:

c_{ij}	travel distance from node i to node j
v	forklift travel speed
f_{ij}	shipment flow (the number of handling units) from i to j
d_i	degree(i) for each $i \in N$. The number of arcs pointing into or out of node i, where $d_i = \sum_{j \in N} f_{ij}$

Variable:

x_{ij}	the numbers of non-required arcs (i, j) used

To determine the value of d_i, the shipment data is pre-processed. For each node i with an origin trailer, d_i is the number of required arcs pointing out of this node. For example, if 5 handling units need to be transferred from node 1, then $d_1 = 5$. For each node i with destination trailer, d_i is the number of required arcs pointing into this node. For example, if 7 handling units are transferred into node 3, then $d_3 = 7$. The problem is formulated as an inter programming model.

Formulation I:

$$\text{Minimize} \quad \sum_{i \in N, j \in N} (c_{ij} / v) x_{ij} \tag{4.12}$$

Subject to:

$$\sum_{j \in N} x_{ij} = d_i \qquad \forall\ i \in N \tag{4.13}$$

$$\sum_{j \in N} x_{ji} = d_i \qquad \forall\ i \in N \tag{4.14}$$

$$\sum_{i \in N'} \sum_{j \in N - N'} x_{ij} \geq 1 \qquad N' \subset N, N' \neq \phi \tag{4.15}$$

$$x_{ij} \geq 0, \qquad \text{integer} \tag{4.16}$$

Since the required arcs on the network *G = (N, A)* must be traveled exactly once by the forklift worker, the objective function minimizes the total travel time for all non-required arcs on the network *G = (N, A)*. Constraint sets (4.13) and (4.14) ensure the graph is balanced. For each node, the number of incoming arcs equals to the number of outgoing arcs. Constraint set (4.15) ensures the graph is connected. The goal is to find a set of non-required arcs to minimize $\sum_{(i,j) \in A} (c_{ij} / v)$. If constraint (4.15) is removed, a Minimum Cost Network Flow Problem (MCNFP) emerges that can be solved using a standard MCNFP algorithm in polynomial time (Ball & Magazine, 1988).

Formulation I determine the set of non-required arcs to minimize the total travel time. The required arcs could also be included in the model formulations as shown in the following integer programming formulation II for single worker FSP. The following indices, parameters and variables are used in the model:

Indices and Sets:

i, j indices of nodes (doors)

N set of all nodes (doors)

R set of required arcs

Parameters:

c_{ij}	travel distance between node i and j
v	forklift travel speed
f_{ij}	shipment flow (number of handling units) from i to j
d_i	degree(i) for each $i \in N$. The number of arcs pointing into or out of node i, where $d_i = \sum_{j \in N} f_{ij}$
u_i	unloading time at node i
l_j	loading time at node j

Variable:

x_{ij}	the numbers of arcs (i, j) used

The integer programming formulation II for a single worker FSP is presented as follows:

Formulation II:

$$\text{Minimize} \quad \sum_{i \in N, j \in N} (c_{ij} / v) x_{ij} + \sum_{(i,j) \in R} (u_i + l_j) x_{ij} \tag{4.17}$$

Subject to:

$$\sum_{j \in N} x_{ij} = d_i \qquad \forall \; i \in N \tag{4.18}$$

$$\sum_{j \in N} x_{ji} = d_i \qquad \forall \; i \in N \tag{4.19}$$

$$x_{ij} = f_{ij} \qquad \forall \; (i,j) \in R \tag{4.20}$$

$$\sum_{i \in N'} \sum_{j \in N - N'} x_{ij} \geq 1 \qquad N \subset N, N' \neq \phi \tag{4.21}$$

$$x_{ij} \geq 0, \qquad \text{integer} \tag{4.22}$$

The difference between formulation I and formulation II is that the required arcs are included in the formulation by constraint (4.20), thus the unloading and loading time for required arcs are included in the objective function (4.17).

The total completion time in the objective function (4.17) includes (1) the travel time for all arcs which is the distance between node i and j (c_{ij}) divided by the forklift travel speed v, and (2) the unloading and loading time for all required arcs. When a work traverses a required arc (i,j), the worker unloads the shipment at node i, resulting in unloading time u_i. The worker then travels to j and loads the shipment at j, resulting in loading time l_j. The objective function (4.17) is to minimize the total completion time for all traveled arcs.

Again, if constraint (4.21) is removed, the remainder of the formulation is a Minimum Cost Network Flow Problem (MCNFP) which can be solved in polynomial time.

Formulation I or II assume that only one worker is required to transfer all the shipments for all origin and destination trailers. Often, multiple dock workers are available to perform the freight transfer jobs during the night in a hub. The following section discusses the modeling approach for the freight unloading and loading sequence for multiple workers (k workers).

4.4 Freight Sequencing Problem for *k* Workers

In hub operations, multiple workers are often used to perform the freight transferring jobs. Given an assignment of trailers to dock doors and the number of workers in hub operations, the freight sequencing problem (FSP) with k workers determines an optimal sequence for each worker to unload shipments from origin trailers and load shipments on the corresponding destination trailers to minimize the maximum completion time of workers during the operations.

With multiple workers perform freight transferring jobs, the total time to transfer freight is dictated by the worker with the largest workload in terms of transfer time. The objective of the FSP with k worker is to minimize the transfer time (make-span) by balancing the workload assigned to workers and reducing the workers to the dock doors.

The FSP with k workers can be described using a directed network, $G = (N, R \subseteq A)$. N is the set of all nodes representing the doors that are assigned to origin and destination trailers. A is the set of all directed arcs (n required arcs plus the link between the ending points of all n arcs). R is the set of all required directed arcs, where R is subset of A. The cost c_{ij} ($c_{ij} \geq 0$) of traversing each arc (i, j) is the distance between i and j. Assume k workers are available. The degree for each node i, d_i can be determined in advance based on the shipment flow data f_{ij} when all trailers are assigned to dock doors. The objective of the FSP with k workers is to find a tour for each worker in the hub operations such that each required arc is traversed once by exactly one worker and the maximum completion time among workers is minimized. The following indices, parameters and variables are used in the model:

Indices and Sets:

i, j	indices for nodes (doors)
k	index for workers
N	set of all nodes (doors)
K	set of all workers
A	set of arcs
R	set of required arcs

Parameters:

c_{ij}	travel distance from node i to j
v	forklift travel speed, constant
f_{ij}	shipment flow (number of handling units) from i to j
d_i	degree(i) for each $i \in N$. The number of arcs pointing into or out of node i, where $d_i = \sum_{j \in N} f_{ij}$
u_i	unloading time at node i, constant
l_j	loading time at node j, constant

Variables:

x_{ij}^{k}	numbers of arcs $(i, j) \in A$ traveled by worker k
z	artificial variable used to minimize the maximum tour cost

The FSP with k workers is modeled as an integer programming:

$$\text{Minimize} \quad z \tag{4.23}$$

Subject to:

$$\sum_{i\in N, j\in N}(c_{ij}/v)x_{ij}^k + \sum_{i\in N, j\in N, (i,j)\in R}(u_i + l_j)x_{ij}^k \le z \qquad \forall \ k\in K \tag{4.24}$$

$$\sum_k \sum_{j\in N} x_{ij}^k = d_i \qquad \forall \ i\in N \tag{4.25}$$

$$\sum_k \sum_{j\in N} x_{ji}^k = d_i \qquad \forall \ i\in N \tag{4.26}$$

$$\sum_{j\in N} x_{ij}^k - \sum_{j\in N} x_{ji}^k = 0 \qquad \forall \ i\in N, \ k\in K \tag{4.27}$$

$$\sum_k x_{ij}^k = f_{ij} \qquad \forall \ (i,j)\in R \tag{4.28}$$

$$\sum_k \sum_{i\in N'} \sum_{j\in N-N'} x_{ij}^k \ge 1 \qquad N' \subset N, N' \ne \phi \tag{4.29}$$

$$x_{ij}^k \ge 0 \qquad \text{integer} \tag{4.30}$$

The objective function (4.23) minimizes the maximum completion time cross all workers. Constraint set (4.24) defines the maximum completion time across all workers including the travel time required for worker *k* to traverse all arcs and the total unloading and loading time for worker *k*. Constraint sets (4.25) and (4.26) assure that each node *i* is visited d_i times. Constraint set (4.27) ensures each worker route is balanced such that the number of non-required arcs by worker *k* into node *i* is equal to the number of required arcs by worker k out of node *i*. Constraint set (4.28) ensures that each required arc is traversed by exactly one worker. Constraint set (4.29) ensures that each worker's tour is connected.

Approximation algorithms based on above IP formulations for the FSP for a single worker and the FSP for *k* workers are developed in Chapter 6.

Chapter V

Solution Approaches for the Trailer-to-Door Assignment Problem

Several heuristic approaches to solve the trailer-to-door assignment problem are presented in this chapter. The heuristics investigated are Pair-wise Exchange, Simulated Annealing, and Genetic Algorithm. An estimator function is used to approximate distances required to transfer all freights from origin trailers to destination trailers. Improvements from solutions developed using heuristics approaches are compared using industry data sets.

5.1 Estimator Function

When a solution is developed for trailer-to-door assignment problem, a trailer-at-a-time freight sequencing approach is assumed such that a worker completes the shipments on a trailer before starting another trailer. In this case, each worker transfers one handling unit from one origin trailer to the destination trailer then returns to the same origin trailer to transfer another handling unit until all handling units on that origin trailer are done. For each handling unit, the worker travels twice the distance from one origin to one destination. When solving the trailer-to-door assignment problem, the assignment of workers to shipments and the sequence for unloading and loading shipments is not known. Therefore the total worker travel distance in the objective function of the trailer-to-door assignment problem is approximated by

$$\text{Total Travel Distance} = \sum_{i}\sum_{j}\sum_{k}\sum_{l} 2 * f_{ij} d_{kl} X_{ik} X_{jl} \tag{5.1}$$

For the last shipment on an origin trailer, therefore the distance to travel from the correct destination trailer to a new origin trailer is estimated by using the distance required to return to the same origin trailer that the worker was unloading.

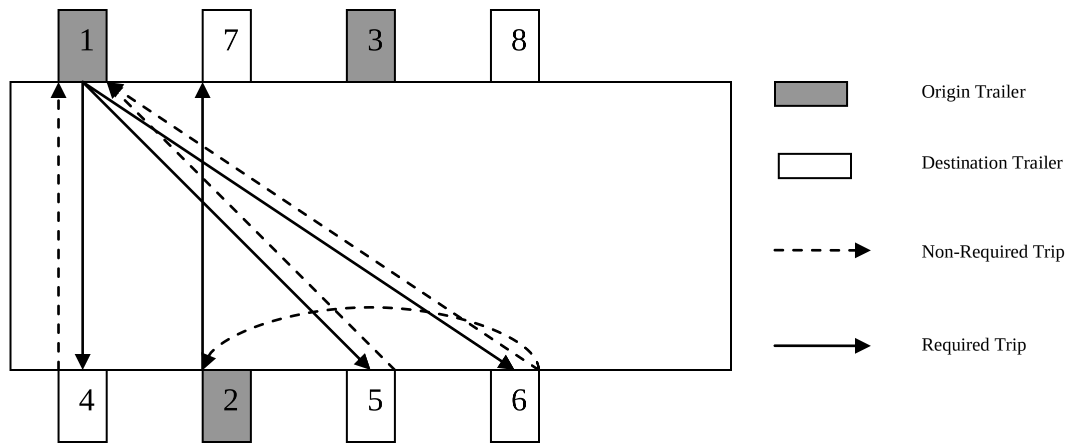


Figure 5. 1 Estimation of Distance for Trailer-to-Door Assignment Problem

For example, assume a worker is assigned to transfer shipments from trailer 1. After transferring the first shipment from trailer 1 to trailer 4, the worker returns to trailer 1 for the second shipment. After the third shipment (last) shipment is transferred from trailer 1 to trailer 6, the worker travels to trailer 2 to start transferring shipments on trailer 2 (rather than returning to trailer 1). Thus the worker's travel distance is $D_{14} + D_{15} + D_{16} + D_{41} + D_{51} + D_{62}$. The approximated total travel distance uses D_{61} instead of D_{62} (since the sequence for unloading and loading is not known and therefore the next trailer is not known). This approximation is also used in the literature (Tsui & Chang, 1990; Tsui & Chang, 1992; and Brown, 2003).

5.2 Approaches for the Trailer-to-Door Assignment Problem

A common approach used in industry for the trailer-to-door assignment problem is a *semi-permanent layout* approach. With a *semi-permanent layout*, destination trailers are assigned to dock doors for a relatively long time (several months). The assignments for destination trailers may be based on geographical proximity. For example, trailers for Roanoke and Charlottesville may be located near each other. The assignment for destination trailers may be also based on historical shipment flow. For example, Atlanta may be located in the dock door with the least distance to all other doors (normally in the middle door of the dock in the "I" shape dock) since the volume of shipments flow to Atlanta is very high. The assignments for origin trailers occur on a nightly basis. Typically dock managers determine these assignments based on actual shipments and their experiences.

In industry hub operations, the destination trailer doors are typically fixed for 3 to 6 months. Since destination doors are fixed relatively, the dock manager only needs to assign origin trailer each day.

Alternatively, a *dynamic layout* may be used for the trailer-to-door assignment problem. With a *dynamic layout*, both origin and destination trailers are assigned to dock doors nightly based on the actual shipment flow. The assignment from *dynamic layout* reflects daily shipment flow characteristics, thus potentially reduces the travel distance of hub workers.

A drawback of a *semi-permanent layout* is that the assignment of destination trailers does not reflect daily shipment characteristics. The total travel distance in the hub operations is generally higher than the total travel distance required for a *dynamic layout*, which use daily shipment data to assign trailers to dock doors. The *dynamic layout*, however, increases the complexity of hub operations by frequently changing the assignments of destination trailers. The comparison of *semi-permanent layout* with *dynamic layout* is provided at the end of this chapter.

To address the trailer-to-door assignment problem, the following approaches are considered:

- semi-permanent layout (with pair-wise exchange);
- dynamic layout (with pair-wise exchange);
- dynamic layout (with Simulated Annealing); and
- dynamic layout (with Genetic Algorithm)

5.3 Semi-permanent Approach

The semi-permanent layout specifies the assignment of destination trailers to dock doors based on the average amount of freight (in historical data) from various origins to destinations. Origin trailers are assigned to dock doors on a nightly basis. In practice, the semi-permanent layout is updated every 3 to 6 months.

To allocate destination trailers to stack doors, in this research, the average number of handling units that flow from an origin to destination trailer is used (called average trailer). These average flow data are from recent months (3 to 6 months) hub operations. The average trailer is generated by taking all the handling units that arrive at the hub over the 3 to 6 months divided by the number of origin trailers that have arrived at the hub. For example, if 500 handling units for a specific destination arrive at the hub during past 3 or 6 months, and there are 25 origin trailers, then the average origin trailer will have 20 handling units for that specific destination. The initial assignment of destination trailers to dock doors is generated randomly. The solution is improved using pair-wise exchange with the objective value of total travel distance based on the average trailer data. After destination trailers are assigned, they are fixed for 3 to 6 months until the assignment is updated with the new average trailer data.

To allocate origin trailers to strip doors, the daily actual flow data between origin trailers and destination trailers is used. Initially the origin trailers are assigned to the strip doors randomly. Then pair-wise exchange procedure is used to improve the solution until the local optimal solution found. Figure 5.2 illustrates the semi-permanent approach with pair-wise exchange for the trailer-to-door assignment problem.

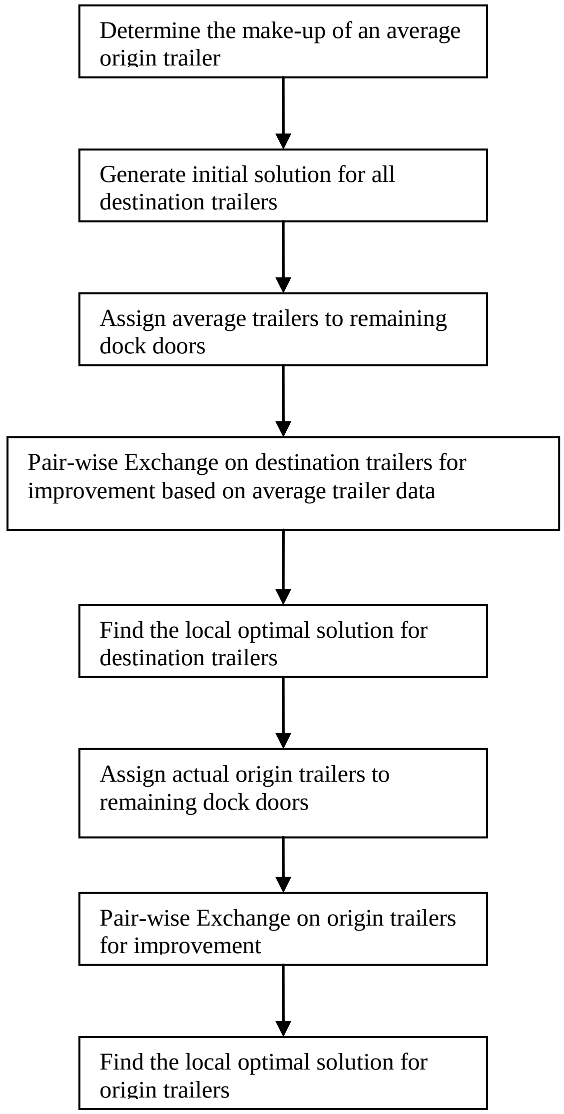


Figure 5.2 Semi-Permanent Trailer-to-Door Assignment Procedure

5.4 Dynamic Approach

The dynamic layout assigns trailers to door each night according to the exact freight that will be flowing through the hub. With a dynamic layout, an initial assignment of origin and destination trailers to dock doors is generated randomly (in this research, the origin trailers are initially placed on the doors on one side of the dock and the destination trailers are initially placed on the doors at the other side of the dock). An improvement process is then applied to assign both origin and destination trailers to the dock doors based on actual shipment flows between origin and destination trailers to minimize total travel distance. The dynamic approach is illustrated in the Figure 5.3. In this research, the improvement process is accomplished using pair-wise exchange, simulated annealing and genetic algorithm, which are described in the following sections.

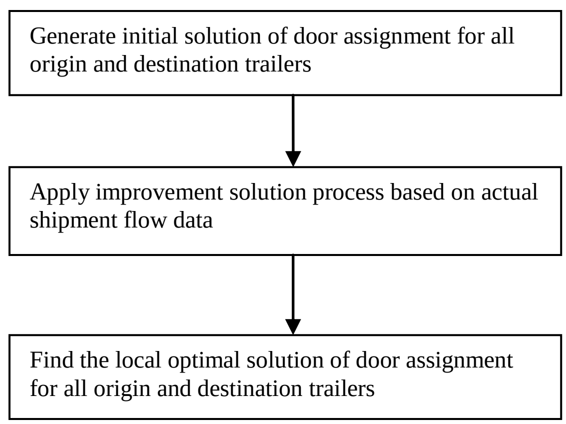


Figure 5.3 Dynamic Trailer-to-Door Assignment Procedure

5.4.1 Local Search and Pair-wise Exchange

Local search algorithms are widely applied to numerous hard computational problems, including problems from computer science, mathematics, operations research, engineering, and bioinformatics. Pair-wise exchange is a commonly employed local search algorithm.

A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution that is defined on the search space. As an example, the neighborhood of a vertex cover is another vertex cover only differing by one node. The same problem may have multiple different neighborhoods defined on it.

Typically, every candidate solution has more than one neighbor solution. The choice of the next move is made using only information about the solutions in the neighborhood of the current one, hence the name local search.

Local search algorithms constitute an interesting class of general approximation algorithms that are based on stepwise improvement of the value of the cost function by exploring neighborhoods. Furthermore, these algorithms have a strong relationship with the Simulated Annealing algorithm, and it is for this reason that we briefly discuss some properties of local search algorithms.

The pair-wise exchange procedure is a widely used local search method in many hard combinatorial optimization problems. For example, in a given initial layout for facility layout problem (an application of QAP), two neighboring facilities are exchanged and the total costs are re-evaluated. If the new layout costs less, then new layout is accepted and exchange between two other facilities are performed. The procedure continues until no improvement occurs.

The pair-wise exchange procedure was first developed for the traveling salesman problem (TSP) in the early 1970s (Lin & Kernighan, 1973). This procedure is often referred to in the literature as the *2-opt* procedure. For the TSP, it involves iteratively removing two edges and replacing these with two different edges that reconnect fragments created by edge removal into a new and more optimal tour. The tour obtained at the end of iteration is always at least as good as the tour obtained at the end of the previous iteration. The iterations stop when there is no improvement found.

The structure of the *solution space* and *neighborhood generation* for the trailer-to-door assignment problem is an important aspect of the pair-wise exchange procedure. For the *solution space*, each valid solution is represented by an n-dimensional vector

$$\underline{x} = (x_1, x_2, x_3, \ldots, x_k, \ldots, x_n)$$

such that for each $i \in [1,n]$, there is a $k \in [1,n]$ such that $x_k = i$. The physical interpretation of such a vector $\underline{x} = (x_1, x_2, x_3, \ldots, x_k, \ldots, x_n)$ is as follows: the first trailer is assigned to door x_1; the second trailer to door x_2; and the n^{th} trailer is assigned to location x_n.

For the *neighborhood generation*, the *solution space* ϕ consisting of all permutations φ is examined to minimize the objective function, using moves from permutation to permutation. A move is a pair-wise exchange or a swapping of two assignments in a given permutation. A permutation reachable within one move is called a neighbor and the neighbor φ' of a permutation φ, obtained by exchanging the assignment of i and k, is thus of the form:

$$\varphi'(i) = \varphi(k),\ \varphi'(k) = \varphi(i)$$

The number of neighbors of a given permutation is $n(n\text{-}1)/2$. It is possible to obtain all permutations moving from neighbor to neighbor given an initial permutation, and thus all of the solution space can be examined within finite time.

For instance, the initial solution vector (1, 2, 3) has a total of 3(2/2) = 3 neighbor solutions, including (2,1,3), (3,2,1), (1,3,2).

In the trailer-to-door assignment problem, an initial assignment is shown in Figure 5.4. The first row represents the trailer identifiers and the second row represents the door identifiers. As shown in Figure 5.5, trailer 1 is assigned to door 66 and trailer 2 is assigned to door 65. To generate a neighborhood solution, an exchange between two adjacent doors is made as shown in Figure 5.5.

Trailer	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Door	**66**	**65**	**64**	**63**	**62**	**61**	**60**	**59**	**58**	**57**	**56**	**55**	**54**

Figure 5.4 Example of a Trailer-to-Door Assignment Solution

Trailer	1	2	3	4	5	6	7	8	9	10	11	12	13
Door	66	65	64	63	62	61	60	59	58	57	56	55	54

Trailer	1	2	3	4	5	6	7	8	9	10	11	12	13
Door	65	66	64	63	62	61	60	59	58	57	56	55	54

Figure 5.5 Exchange of Two Door Assignment

In the trailer-to-door assignment problem with *n* doors and *n* trailers, each of the doors can be inserted in (*n-1*) other positions for total possible pair-wise exchanges of *n*(*n-1)/2*. The complete solution space is *n*!

Like other local search techniques, the pair-wise exchange procedure may stop before an optimal solution is found and the optimal solution may lie far from the neighborhood of the solutions crossed by the algorithm. To minimize the likelihood of getting trapped in a local optima, simulated annealing and other meta-heuristics are widely used to eliminate the disadvantage in the local search.

The flow chart of pair-wise exchange is summarized in Figure 5.7.

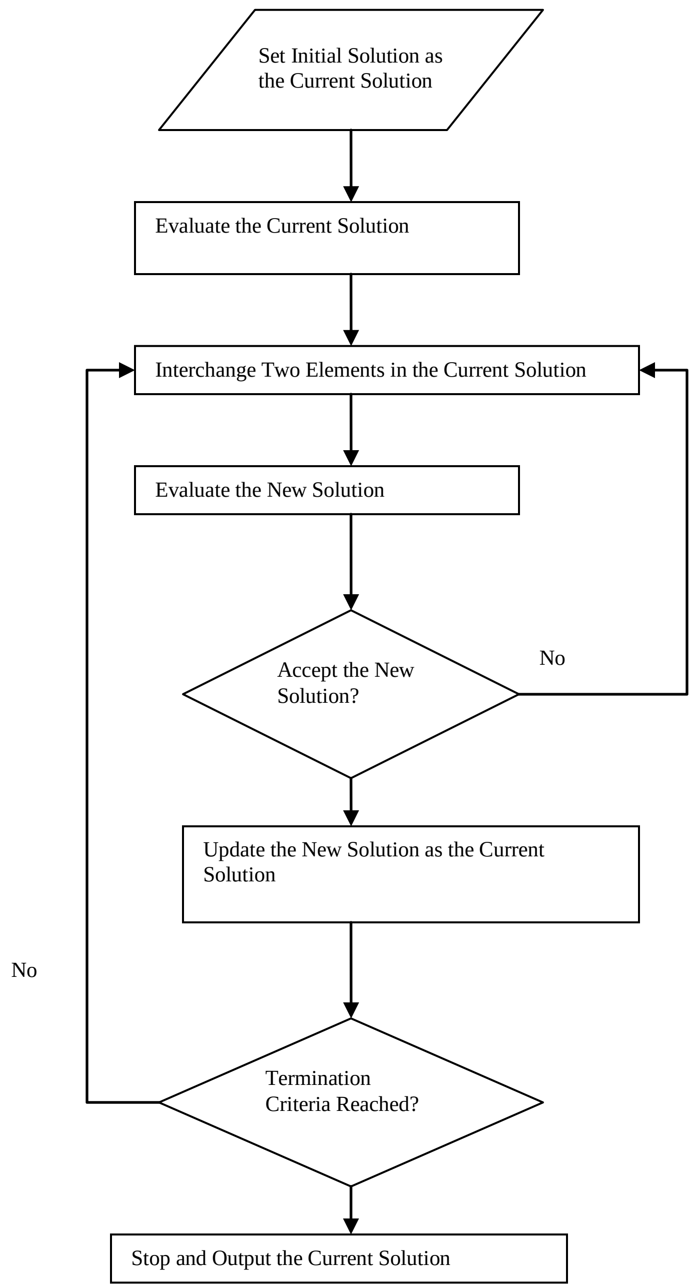


Figure 5.6 Pair-wise Exchange Procedure Flow Chart

The pair-wise exchange algorithm is implemented as follows for the trailer-to-door assignment problem:

Step 0: Form an initial assignment of trailers to doors as shown in Figure 5.1.

Step 1: Calculate the total distance for the initial solution; let the initial solution be the current solution;

Step 2: Exchange the door assignment of trailer i (i = 1, 2, …, n) to the door assignment of trailer j ($j = i$ +1 and $j \leq n$).

Calculate the new total distance;

If the new total distance < the current total distance,

accept the new assignment solution as the current solution;

update the current total distance;

Else does not exchange the assignment;

Step 3: Increment j and go to step 2;

Step 4: Increment i and go to step 2;

Step 5: Output the current solution.

The above pair-wise exchange algorithm is implemented in C++ to solve the trailer-to-door assignment problem. The results are compared to other heuristics.

5.4.2 Simulated Annealing

Simulated annealing (SA) belongs to a general class of "probabilistic meta-algorithms" used to approximate global optimal in complex combinatorial optimization problems. SA was invented by Kirkpatrick, Gelatt and Vecchi (1983).

A predominant conceptualization of SA in the literature is "by following the analogy between finding minimum energy states in a physical system and finding minimum cost configurations in a combinatorial optimization problem" (Vidal, 1993). SA can be described as a process "…which first 'melts' the system being optimized at a high effective 'temperature', then (slowly) lowers the 'temperature' in stages until the system 'freezes' and no further changes occur. Its objective is finding a configuration (or state) for which a certain cost function takes its minimum value" (Daganzo, 2005).

The main advantage of an SA algorithm is that it attempts to avoid becoming trapped in a local optimal by sometimes accepting transitions corresponding to an increase in cost. In contrast, local search algorithms only accept transitions corresponding to decease in cost (minimization problem).

The SA algorithm generates new configurations with some probabilistic rules, and either accepts or rejects them depending on their relative cost. The probability of acceptance is controlled by a parameter, *T*, analogous to the temperature in physics.

If the algorithm is in a certain configuration at time *t*, the function randomly selects a new configuration from a set of feasible neighbor configurations. The most common accept function is based on Boltzmann's probability density distribution. If the cost of the new configuration, $c(s_{t+1})$ is less than the cost of the old configuration, $c(s_t)$, the new configuration is accepted with probability one. Otherwise, it is accepted with probability:

$$e^{\{-\frac{[c(s_{t+1})-c(s_t)]}{T_t}\}} . \tag{5.2}$$

The time *t* is increased by one unit regardless.

Let (*S, c*) be an instance of a combinational optimization problem and *i* and *j* two solutions with cost *c(i)* and *c(j)*, respectively. Then the acceptance criterion determines whether *j* is accepted from *i* by applying the following acceptance probability:

$$\mathrm{P}_T\,\{\text{accept } j\} = \begin{cases} 1 & c(j) \le c(i) \\ e^{\{\frac{[c(i)-c(j)]}{T}\}} & c(j) > c(i) \end{cases} \tag{5.3}$$

where $T \in \Re^+$ denotes the control parameter.

The flow chart for simulated annealing is summarized in Figure 5.9:

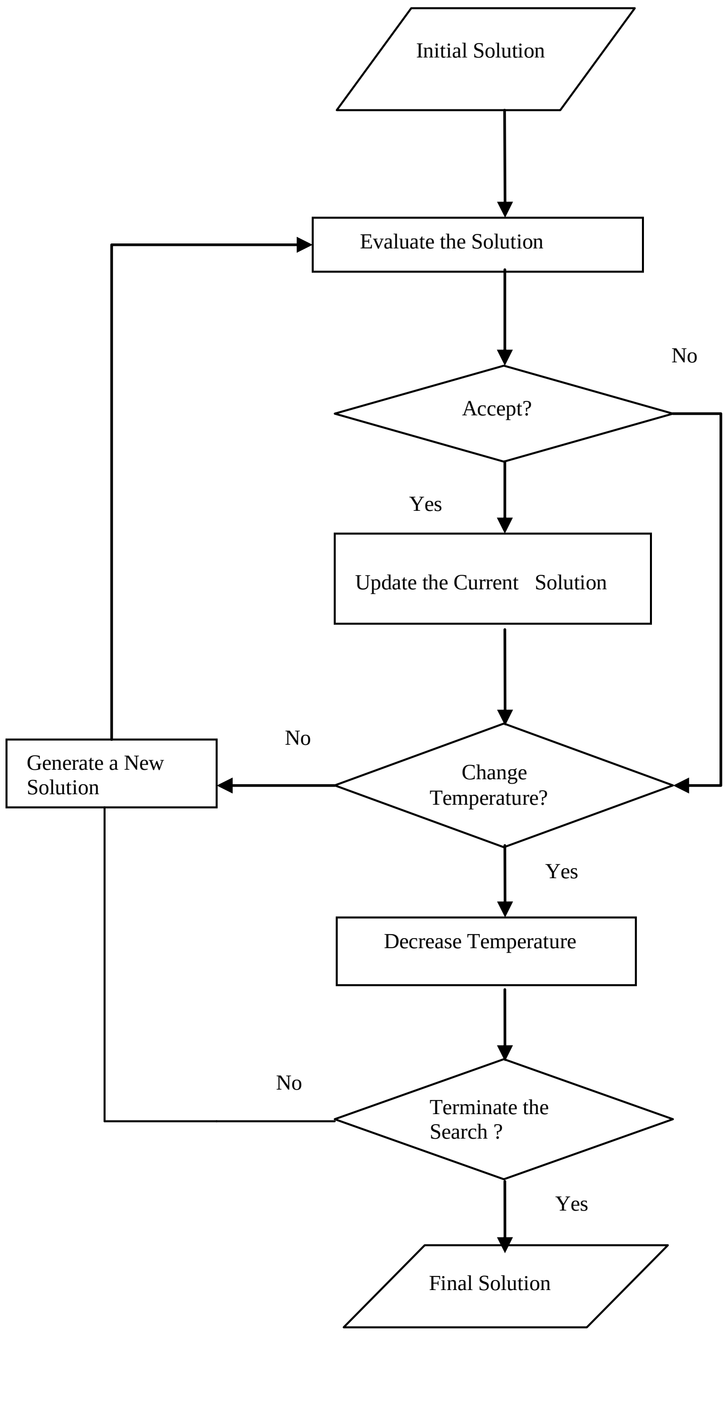


Figure 5.7 Simulated Annealing Procedure Flow Chart

The update function defines the ‘annealing schedule” which reduces the temperature from T_t to $T_{t+1.}$ Initially, T_t is set to a high value and it is decreased at each step until T_n is zero. In this way, the system is expected to wander initially towards a broad region of the search space containing good solution, ignoring small features of the energy function. Then the approach drifts toward low-energy regions that become narrower and narrower.

Since a SA algorithm is not guaranteed to converge in polynomial time, values for the following parameters are specified: a) a finite number of transitions at each value of T, and b) a finite sequence of values of T.

Implementations of SA are problem specific. To solve a particular combinatorial optimization problem by SA algorithm, a number of decisions have to be made. These decisions are divided into generic and problem specific decisions. The generic decisions apply to any implementation of SA and relate to designing the cooling scheme (setting initial temperature, changing temperatures, stopping criteria, etc.). The problem specific decisions are closely related to the actual problem to be solved. These decisions include generating an initial solution and neighborhood solutions, and evaluating the change of objective function.

An effective application of simulated annealing includes: 1) an efficient method to calculate the cost of a configuration, and 2) an efficient method of altering the system. For example, for the classical TSP problem, the cost function is the tour length. The SA algorithm then tries variations on this tour and checks for improvements in tour length.

The implementation of SA to solve the trailer-to-door problem is presented in the following sections. The techniques to generate an initial solution, generate neighborhood solutions, evaluate neighborhood solution, and design the annealing scheme are described.

5.4.2.1 Generate Initial Solution

For the trailer-to-door assignment problem, a feasible solution is represented as an array (as in the pair-wise exchange approach), with the value of the row representing a physical door at the hub and each column index representing a trailer. For example, in Figure 5.1, element 4 has value of 63, which means that trailer 4 is assigned to door 63. Assume trailers are indexed 1 to 13 (including both origin trailers and destination trailers). Trailers are assigned in the sequences to the doors with the largest index. For example, if 66 doors are available in the hub during the operations and 13 origin and destination trailers need to be assigned, then the trailer 1 will be assigned to door 66 and trailer 2 will be assigned to door 65.

This representation of a trailer-to-door assignment solution is feasible since it adheres to the assignment constraints of the underlying QAP where a trailer is only assigned to one door and a door has at most one trailer.

The cost function of this feasible assignment solution is evaluated by the travel distance required to unload and load all trailers. Assuming the trailer-at-a-time approach for unloading and loading shipment, the travel distance required by a worker is calculated by the estimator function described in section 5.1.

5.4.2.2 Generate Neighborhood Solution

The neighborhood solution of a feasible solution can be generated by exchanging any pair of assignments of doors. For example, the two solutions in Figure 5.2 are neighborhood solutions. With n trailers, $[n(n-1)/2]$ neighborhood solutions exist for each trailer assignment. So a pair-wise exchange approach is used to exchange one trailer door assignment to the other of $(n-1)$ positions to obtain the neighborhood solutions. The potential pair-wise exchanges are examined in the order (1,2), (1,3),…,(1,n), (2,3),…, (n-1, n).

As described by Rainer and Offermann (1997), this cyclic generation of neighborhood solutions is preferred to the random generation of neighborhood solutions. With random generations, some uphill move may be accepted before the local optimum is sampled and thus the local optimum may never be reached. In cyclic generation of neighborhood solution (like the pair-wise exchange approach), all neighborhoods are tried once before any are considered for a second time. Pair-wise exchange can also decrease the computation time compared to random generation of neighborhood solutions.

5.4.2.3 Evaluate Neighborhood Solution

After the neighborhood solution is generated, the cost of the solution is calculated and compared to the cost of the old solution. If the new cost is less than the old cost, the new neighborhood solution is accepted and replaced for the old solution. If the new cost is greater than the old cost, the new neighborhood solution is accepted with the following probability:

$$e^{(-(\text{new cost} - \text{old cost})/T)}$$

where T is a parameter which controls the accepting probability.

5.4.2.4 Annealing Scheme

The annealing scheme is used to reduce the temperature of the system gradually. The stages of initial temperature, change in temperature, decrease in temperature, and stopping criteria are considered in this section.

The *initial temperature* is set to some large value, thereby permitting almost all attempted moves. The temperature is gradually lowered in a predetermined fashion until no further change seemed likely and the system is considered to be “frozen.” In the trailer-to-door assignment problem, the initial temperature is set as follows:

$$T_0 = Z_0 / I \tag{5.4}$$

where I is the temperature divisor, Z_0 is the cost of initial solution. If the new cost increases by 10%, 5% and 1% from the old cost, for I = 40, the acceptance probability α is 0.018, 0.135 and 0.368 respectively. If the algorithm starts from initial solution (the solution from pair-wise exchange algorithm), the acceptance probability α is not set very high because a high temperature might destroy the good solution search. If the algorithm starts from a random initial solution, the acceptance probability α is set to be high to allow more random moves at the high temperature for a “bad” solution. More details on choosing parameters such as T_0, I and α are provided in Section 5.4.2.5.

For *change in temperature*, two conditions from the literature (Reeves, 1993) are used to change the temperature if one of the following is satisfied:

1. Accepted 15*N perturbations, or
2. Attempted 100*N perturbations

where N indicates the number of degrees of freedom of the problem (the total number of trailers during the hub operations). For example, if a total of 30 origin and destinations are in the hub operations, the temperature in the annealing scheme is changed either if there are 15*30 = 450 new neighborhood solutions accepted or there are 100*30 = 3000 new neighborhood solutions attempted.

For *decrease in temperature*, the geometric law is adopted to carry out the temperature change:

$$T_{k+1} = \alpha * T_{k}, \ \alpha \in [0,1] \tag{5.5}$$

The geometric law is easy to implement in trailer-to-door assignment problem. Setting the value of α will be discussed in the next section.

Two *stopping criteria* are used to determine when to terminate the search process:

1. Temperature is low (near zero), or
2. Several successive temperature stages without any acceptance

Both criteria are included in this research.

5.4.2.5 Simulated Annealing Parameters Experiment Design

With the simulated annealing (SA) algorithm, significant benefits can be gained by determining preferred values for certain search parameters. In this section, experiments are performed to fine-tune the SA algorithm to find the best results for dock door assignment problem.

5.4.2.5.1 Parameters Settings

The following four parameters of the simulated annealing (SA) algorithm are considered to fine-tune the SA algorithm:

Initial Temperature Divisor (I);

Temperature Change Factor (α);

Number of Acceptance for New Solution (N_a), and

Number of Attempt for New Solution (N_t).

The initial temperature highly influences SA algorithms' success. High initial temperatures lead to relatively high probability of accepting increased objective cost function and helping the search process leave a local minimum. In our research, initial temperature t_1 is determined by

$$t_1 = Z_0 / I \tag{5.6}$$

where Z_0 denotes the initial objective function value. The probability of accepting increased objective cost function is expressed as follows:

$$\text{Prob}\ (\Delta C) = e^{(-\Delta C/t1)} \tag{5.7}$$

The Prob (ΔC) can be calculated by using $\Delta C/Z_0$ and different I values. Table 5.1 summarize the probabilities with different I values and different ratios of $\Delta C/Z_0$.

For example, if I = 40, the probabilities to accept an increased cost function is 0.368 for a solution 1% above the initial objective function value, and the probabilities are 0.135 for 5% deviations. As shown, as the value of I increases, the probabilities to accept increased cost function are decreasing.

Table 5.1 Initial Temperature Divisor *I* and Probabilities to Accept Increased Cost Function

I	**$\Delta C/Z_0$ = 1%**	**$\Delta C/Z_0$ = 5%**	**$\Delta C/Z_0$ = 10%**
I = 40	0.368	0.135	0.018
I = 160	0.202	0.0003	0.0000001
I = 200	0.135	0.000045	0
I = 400	0.018	0	0

The Temperature Change Factor, α, determines the cooling schedule of SA. In the research a power cooling schedule is used for temperature reduction, which is $T_k+1 = \alpha T_k$ with $0 < \alpha < 1$. In this experiment α is varied between 0.5 and 0.9.

The Number of Acceptances for New Solution (N_a) and the Number of Attempts for New Solution (N_t) determine the equilibrium for each temperature t_i. At each temperature, the inner loop of the searching process will terminate when the number of iterations N_a or N_t exceeds the specified numbers. In this experiment, those values are specified as 500 and 3000.

5.4.2.5.2 Experimental Design

A two-level factorial design approach is used to test the effects of SA parameters. A two-level full factorial design is often abbreviated as 2k, where k is the number of parameters. In a two-level factorial design, each parameter is tested at a low level and high level value. In this experiment, with four parameters, there are a total of 16 (2^4) experiments are conducted. Table 5.2 summarizes the low and high levels value for each SA parameters. Table 5.3 summarizes the 16 experiments for the combinations of parameters with low or high values. The low level for a parameter is denoted by “-,” and the high level value for a parameter is denoted by “+.”

Table 5.2 Parameters and Levels Considered in the Experiments

Parameter	Low (-)	High (+)
Initial Temperature Divisor (I)	40	400
Temperature Change Factor (α)	0.5	0.9
Number of Acceptances (N_a)	500	3000
Number of Attempts (N_t)	500	3000

The low value of I is chosen from Meller and Bozer (1996) for the SA algorithm used for the facility layout problem which is similar to the QAP. The high value of I is set to 400 to represent a low probability of accepting a solution with increased cost function.

The low value of α is set to 0.5 to test the effect of temperatures changing more quickly. The high value of α is set to 0.9 to test the effect of temperatures changing more slowly.

The values for the Number of Acceptances (N_a) and the Number of Attempts (N_t) evaluate whether we need more or less permutations in each loop to find the best solution. In this experiment, these values are specified as 500 and 3000.

Table 5.3 Combination of Parameters Settings

Experiment	Combination of Parameters
1	I-, α-, N_a-, N_t-
2	I-, α-, N_a-, N_t+
3	I-, α-, N_a+, N_t-
4	I-, α-, N_a+, N_t+
5	I-, α+, N_a-, N_t-
6	I-, α+, N_a-, N_t+
7	I-, α+, N_a+, N_t-
8	I-, α+, N_a+, N_t+
9	I+, α-, N_a-, N_t-
10	I+, α-, N_a-, N_t+
11	I+, α-, N_a+, N_t-
12	I+, α-, N_a+, N_t+

13	$I+$, $\alpha+$, N_a-, N_t-
14	$I+$, $\alpha+$, N_a-, N_t+
15	$I+$, $\alpha+$, N_a+, N_t-
16	$I+$, $\alpha+$, N_a+, N_t+

5.4.2.5.3 Experimental Results

Seven data sets from the hub operations of two LTL carriers are used to perform the 16 experiments to determine the preferred combinations of four SA parameters. The data sets are summarized in Table 5.4.

Table 5.4 Data Sets for Experiments

Data Set	Origin Trailers	Destination Trailers	Number of Shipments	Dock Doors
1	6	2	11	8
2	9	2	23	11
3	16	15	178	31
4	16	15	178	31
5	16	16	201	32
6	17	17	173	34
7	62	33	652	95

Each data set includes two data file: the dock door distance and the shipment flow data. The first file includes the distances between all dock doors. The second file includes the shipments data to be handled in the hub operations. The shipment data contains the origin, the destination and the handling units for each shipment.

The objective function value for each data sets and each experiment using SA algorithm is summarized in Table 5.5. The computational time for each data sets and each experiment using SA algorithm is summarized in Table 5.6.

Table 5.5 Objective Values of SA algorithm for 16 Experiments with 7 Data Sets

	Data Sets						
Experiment	1	2	3	4	5	6	7
1	**2818***	5416	101532	103468	80004	46980	501974
2	**2818***	5416	100860	101622	81982	45356	467402
3	**2818***	5416	101532	103468	80004	46980	501974
4	**2818***	5416	100860	101622	81982	45356	467402
5	**2818***	5416	100534	**99954***	80524	**44892***	**461790***
6	2840	**5308***	**100284***	102986	**79072***	45042	469320
7	**2818***	5416	100534	**99954***	80524	**44892***	**461790***
8	2840	**5308***	**100284***	102986	**79072***	45042	469320
9	2862	5614	102856	104770	82966	47026	484798
10	2862	5416	102532	103468	81740	45824	483846
11	2862	5614	102856	104770	82966	47026	484798
12	2862	5416	102532	103468	81740	45824	483846
13	2840	5570	101918	100920	81142	44974	484798
14	2840	5416	103118	105314	81770	45348	483846
15	2840	5570	101918	100920	81142	44974	484798
16	2840	5416	103118	105314	81770	45348	483846

* indicate minimum value across 16 experiments

Table 5. 6 Computational Time of SA algorithm for 16 Experiments with 7 Data Sets

	Data Sets						
Experiment	1	2	3	4	5	6	7
1	1	1	1	1	1	1	13
2	1	3	3	4	4	6	14
3	1	1	1	1	1	2	11
4	1	4	3	4	4	6	15
5	1	4	5	5	5	8	59
6	1	4	19	13	12	40	49
7	1	4	5	6	6	7	28
8	10	4	17	25	23	36	22
9	1	1	1	1	1	1	8
10	1	1	1	3	2	4	8
11	1	1	1	1	1	1	8
12	1	1	1	3	2	4	3
13	1	1	3	5	4	8	14
14	11	6	17	15	26	24	14
15	2	1	4	4	4	4	12
16	10	6	13	14	12	24	15

As shown in Table 5.5, the minimum objective value is found in four out of the seven data sets in experiments 5 and 7. Also the minimum objective value is found in three out of the seven data sets in experiments 6 and 8. Experiments 5, 6, 7 and 8 had two common parameters: the initial temperature divisor I = 40 and the temperature change factor α = 0.9. As shown in Table 5.6, the average objective values for I- and α+ are lower than the average object values for I+ and α-. The results imply that low level Initial Temperature Divisor (with relatively high probabilities of accepting increased cost function value) and high level Temperature Change Factor (slowly reduce the temperature) offer SA more opportunities to reach the best solution. Table 5.5 also shows that the computational time for experiment 5, 6, 7 and 8 are much higher than the computational time for the rest of experiments. The high probability of accepting increased cost function and the slow temperature reduction rate tend to take more time to reach the final convergence. Thus, a trade off occurs between obtaining best solution and reducing computational time. The computational times for experiment 5, 6, 7 and 8, however, are within 60 minutes for seven data sets. This is still a reasonable computational time to obtain the best solution for the trailer-to-door assignment problem.

Table 5.7 Average Objective Values of SA algorithm for 4 parameters with 7 Data Sets

	1	2	3	4	5	6	7
I-	2825	5466	101907	103221	81447	45793	482283
I+	2851	5504	102606	103618	81905	45793	484322
α-	2840	5466	101945	103332	81673	46297	484505
α+	2840	5467	101857	103073	81145	45121	479321
N_a-	2840	5466	101907	103221	81447	45793	482283
N_a+	2840	5466	101907	103221	81447	45793	482283
N_t-	2840	5504	101710	103053	81159	46327	490523
N_t+	2840	5416	102170	103348	81831	45393	476104

For the Number of Acceptances (N_a) and the Number of Attempts (N_t), there is no difference between the results obtained from the experiments with the low and high values for those two factors. From Table 5.7, the average objective values obtained from N_a-, N_a+, N_t-, and N_t+ are all the same.

5.4.3 Genetic Algorithms

Genetic algorithms are a particular class of evolutionary algorithms used to find true or approximate solutions to optimization and search problems. Genetic algorithms, proposed by Holland (1975) and popularized by Goldberg (1989), are based on the mechanics of natural selection and genetics in biology. Similar to simulated annealing, a genetic algorithm (GA) uses a "guided probabilistic procedure" that uses a population of solutions and employs a survival of the fittest strategy.

GAs creates an initial string of population of "individuals," each representing a possible solution to a given problem. Each individual is assigned a "fitness score" according to how good of a solution to the problem it is. The highly-fit individuals are given opportunities to "reproduce," by "cross breeding" with other individuals in the population. This produces new individuals as "offspring," which share some features taken from each "parent." The least fit members of the population are less likely to get selected for reproduction, and so "die out." A whole population of possible solutions is produced by selecting the best individuals from the current "generation," and mating them to produce a new set of individuals. This new generation contains a higher proportion of the characteristics possessed by the good members of the previous generation. In this way, over many generations, good characteristics are spread throughout the population. By favoring the mating of more fit individuals, the most promising areas of the search space are explored. If the GA has been designed well, the population will converge to an optimal solution to the problem.

A population of solutions is as an array of bits and fitness can be defined by "fitness functions" to evaluate the solution. For example, in the Knapsack problem, a population of solution is an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such population is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of all objects in the knapsack if the population is valid or 0 otherwise.

A simple genetic algorithm processes a finite population of fixed-length binary strings and consists of three operators:

- Selection,
- Crossover, and
- Mutation.

Selection ensures survival of the fittest within the Genetic Algorithm. The key notion is to give preference to better individuals. Individual solutions are selected through fitness-based process, where more fit solutions (as measured by a fitness function) are typically more likely to be selected. This evaluation can come from a formal objective function, or it can come from the subjective judgment of a human observer. On the other hand, most fitness functions are stochastic and designed so that a small proportion of less fit solutions are selected. This approach tends to keep the diversity of the population large, preventing premature convergence on poor lotions.

Crossover proceeds in the following steps. Two individuals are chosen from the population by using the selection operator then a cross site along the string length of two individuals is chosen. Then position values are exchanged between the two strings following the cross site. In this way, a new solution is created which typically shares many of the characteristics of its "parents." New parents are selected for each child and the process continues until a new population of solutions of appropriate size is generated.

Mutation is the occasional (low probability) alteration of a bit position. When used together with selection and crossover, mutation acts both as an insurance policy against losing needed diversity and as a hill-climbing algorithm.

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies optimal criteria;
- Fixed number of generations reached;
- Allocated budget of computation time or money reached;
- The fitness of the highest ranking solution is reaching or has reached a plateau such that successive iterations no longer produce better results;
- Manual inspection; and

- Combinations of the above.

The key to improving the GA is to reduce the time needed to calculate the fitness. From GA procedures, the fitness value for the same chromosomes is recalculated repeatedly. If previous calculation of fitness value can be efficiently saved, computation time will be reduced significantly. By efficiently storing the fitness value, GA can be dramatically improved.

5.4.3.1 Genetic Algorithms and QAPs

Several GA approaches have been used in solving the QAP. Kochhar et al. (1998) apply GAs to the facility layout problem (a version of the quadratic assignment problem). They find solutions within 1% to 5% of the best known solutions. Tavakkoli-moghaddain and Shayan (1994) also use GA to address the facility layout problem. They state "GAs have successfully been applied to NP hard problems such as those resulted in mathematical modeling of facilities design problems." Their GA provides results that bettered previous solution values obtained using a branch-and-bound technique by up to 11.8%. Ahuja et al. (1995) obtained promising studies on large scale QAPs using a greedy Genetic Algorithm. The greedy GA combines techniques from GRASP algorithms and GAs. The greedy GA is tested on all QAPLIB problems, obtaining the best-known solutions for 103 out of 132 test problems (Ahuja, et al., 1995).

.

5.4.3.2 Genetic Approach for the Trailer-to-door Assignment Problem

For the trailer-to-door assignment problem, each individual in the GA is encoded to define a solution. The representation of solution is similar to the representation of solution in the pair-wise exchange and simulated annealing method (Figure 5.1). The crossover procedure for trailer-to-door assignment problem is shown in Figure 5.11 and Figure 5.12.

Trailer	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Door	66	65	64	63	**62**	**61**	60	**59**	58	**57**	56	55	**54**
Door	58	63	56	**57**	54	**59**	66	**62**	**61**	55	64	65	60

Figure 5.8 Trailers to Doors Assignment Before Crossover

Suppose that two individual assignments are selected to mate. The crossover of genes is performed as follows. First, randomly select a door swap window. Second, identify genes within the swap window that are common to the two parents. In this case, swap window from door 4 to door 11 is chosen. The common genes within the swap window [4,11] are ordered (62, 61, 59, 57) and (57, 59, 62, 61), respectively. After the crossover, the resulting two offspring solutions are shown in Figure 5.12. These new solutions of the door assignments are evaluated by re-calculating the objective function.

Trailer	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Door	66	65	64	63	**57**	**59**	60	**62**	58	**61**	56	55	**54**
Door	63	56	**62**	54	**61**	66	**59**	**57**	55	64	65	60	63

Figure 5. 9 Trailer- to-Door Assignment after Crossover

For this research, the GA approach available through the Excel Premium Solver Evolutionary Algorithm is used to change the assignment solutions in search of the optimal solution. The solutions from this evolutionary algorithm are compared to other heuristics in the next section.

5.5 Case Study for Four Solution Approaches for the Trailer-to-Door Assignment

Using the data from two LTL carriers, the following approaches are compared:

- Semi-permanent layout with pair-wise exchange (SPW)
- Dynamic layout with pair-wise exchange (DPW)
- Dynamic layout with Simulated Annealing (DSA)
- Dynamic layout with Genetic Algorithm (DGA)

The size of the problems varies from 8 to 95 trailers (including origin trailers and destination trailers). The total number of shipments on origin trailers to be transferred to the destination trailers varies from 11 to 652. Each data set represents one week night of hub operations. The data sets are summarized in Table 5.4.

Pair-wise exchange and simulated annealing are implemented in C++ using Microsoft Visual Studio 2005. The Genetic Algorithm is implemented using the EXCEL premium solver (evolutionary algorithm). The programs are run on an IBM ThinkPad T43 machine (1.73GHz, 0.99 GB RAM). The total travel distance results are summarized in Table 5.8. As shown, the dynamic layout approaches outperform the semi-permanent layout approach for all data sets.

Table 5.8 Total Travel Distance for Four Approaches

Data Set	SPW (ft.)	DPW (ft.)	DSA (ft.)	DGA (ft.)
1	4954	3226	2818	2818
2	8569	5570	5308	5308
3	161870	106918	100284	101416
4	162266	103720	99954	103160
5	107300	80962	79072	81304
6	60836	46726	44892	46936
7	900094	525640	461790	525960

As shown in Table 5.9, the improvement of DPW over SPW ranges from 23% to 42%. The improvement of DGA over SPW ranges from 23% to 43%. The improvement of DSA over SPW ranges from 26% to 49%. These results demonstrate the potential improvement for transitioning to a dynamic layout. The significant reduction on the travel distance during hub operations provides LTL carriers the opportunities to reduce the time to transfer all shipments and also reduce the labor costs for the hub operations.

Table 5. 9 Total Travel Distance Improvements from Dynamic Approaches

Data Set	DPW-SPW %	DGA-SPW %	DSA-SPW %
1	35%	43%	43%
2	35%	38%	38%
3	34%	37%	38%
4	36%	36%	38%
5	25%	24%	26%
6	23%	23%	26%
7	42%	42%	49%

For the dynamic layout approaches, the simulated annealing approach (DSA) performs the best among three heuristics. As shown in Table 5.9, the DSA out performances the DPW for all data sets (with an improvement of 2% to 13%). For smaller size problem both DSA and DGA found optimal solutions. As the problem size increases, DSA out performances DGA (with an improvement of 1% to 12%).

As shown in Table 5.10, the SPW and DPW generate solutions within seconds for all data sets. The computation time of DSA increases as the problem size increases. The computational times for medium size problem (31 to 34 doors) are within 20 minutes for each case. The computational time for large size problem (95 doors) is within 30 minutes. Most of the computational time is consumed at the low temperatures where the search progress is less productive than it is at the higher temperature. The DGA solutions are generated within approximately 0.11 to 10 minutes.

Table 5.10 Computational Time for Four Approaches

Data Set	SPW (min)	DPW (min)	DGA (min)	DSA (min)
1	0.1	0.1	0.1	1
2	0.1	0.1	0.1	4
3	0.1	0.1	1	7
4	0.1	0.1	1	8
5	0.1	0.1	1	12
6	0.1	0.1	1	7
7	0.1	2	10	28

5.6 Summary of Solution Approaches for Trailer-to-Door Assignment Problem

Several heuristics are explored in this chapter to compare the *semi-permanent layout* and *dynamic layout* for the trailer-to-door assignment problem. Using the data from two LTL carriers, the following approaches are compared:

- Semi-permanent layout with pair-wise exchange (SPW)
- Dynamic layout with pair-wise exchange (DPW)
- Dynamic layout with Simulated Annealing (DSA), and
- Dynamic layout with Genetic Algorithm (DGA).

For the data sets considered, the dynamic layout with the simulated annealing algorithm (DSA) provides the best solution in terms of total travel distance for all test data sets. The dynamic approach with pair-wise exchange (DPW) and genetic algorithm (DGA) also outperform the semi-permanent approach with pair-wise exchange (SPW) in all test data sets with short computational time. These results demonstrate the substantial improvements from using a dynamic layout for reducing total travel distance during the hub operations.

To implement a dynamic layout in the hub operations, however, the following two issues need to be considered: the cost of potential misdirected freight as workers adjust to the new assignment every night and the dock management of creating new dock door assignments every night. For the first factor, due to the advances in scanner technology, the cost of misdirected freight can be avoided if dock workers scan freight at the destination trailer (Gue, 1995). For the second factor, an advanced dock door management system with embedded dynamic layout approach can be provided for each hub of an LTL carrier. This dock door management system can be used each day before the hub operations begin, with the shipment flow data transited from each customer freight collection center. With the recent advances of information technology, LTL hubs are in the position to use dynamic layout in the near future.

Chapter VI

Solution Approaches for the Freight Sequencing Problem

This chapter describes heuristic approaches to solve the Freight Sequencing Problem (FSP) which is modeled as a directed Rural Postman Problem. The heuristics for the FSP for a single worker are presented followed by the heuristics for the FSP for multiple workers.

6.1 Solution Approaches for the FSP for a Single Worker

In this section, the Balance-and-Connect Algorithm (BCA) for the FSP for a single worker is developed and compared to the Nearest Neighbor algorithm and the trailer-at-a-time approach. A case study with data sets is presented to demonstrate the effectiveness on the BCA.

6.1.1 The Balance-and-Connect Algorithm for the FSP

The Balance-and-Connect Algorithm, based on formulation (4.17) - (4.21), is adapted from a framework presented by Ball and Magazine (1988) for a similar problem in electronic assembly systems. In the mathematical model of the FSP, constraint (4.21) assures that the network is connected. If this constraint is removed, a minimum cost flow problem remains that can be solved using a network flow algorithm. After the minimum cost flow problem is solved to generate a balanced network, several disconnected sub tours may exist. If sub-tours are found, a minimum spanning tree algorithm can be used to generate a connected graph which contains an Euler tour. Using an Euler tour finding procedure, an Euler tour is generated to correspond to the worker's sequence of unloading and loading shipments. Since the balanced network is found first and then the network is connected if sub tours exist, the procedure is referred to as the Balance-and-Connect Algorithm. This sequential approach will produce an approximate solution to the FSP that may not necessarily be optimal for the original problem since it does not address the interaction between the balancing and connecting steps.

The Balance-and-Connect Algorithm is summarized as follows:

Step 1: Setup the directed RPP network $G_R = (N, R)$;
Step 2: Determine the degree for each node on the network;

Step 3: Solve the relaxed problem;

Step 4: Check the connectivity of graph $G = (N, A)$**;** If G is not connected, then go to Step 5; otherwise go to step 6;

Step 5: Connect the sub tour in the graph $G = (N, A)$**;**

Step 6: Form a network that consists of all required arcs and non-required arcs.

Step 7: Apply the Euler tour algorithm to find a sequence.

Details of these steps are discussed in the following:

Step 1: Setup the directed RPP network

Set up a directed RPP network $G_R = (N, R)$ where $N = \{1, 2, \ldots\ i, \ldots n\}$ is the set of doors with assigned origin and destination trailer and R is the set of all required arcs. The assignment of origin and destination trailers to dock doors is based on the solution of trailer-to-door assignment problem.

Step 2: Determine the degree for each node on the network;

Compute d_i = degree(i) for all nodes i in the graph $G_R = (N, R)$. For each node i with an origin trailer, d_i is the number of required arcs pointing out of this node. For each node i with a destination trailer, d_i = degree(i) is the number of required arcs pointing into the node. The number of required arcs is the number handling units pointing into or out of that node. For example, if there are 5 handling units that need to be transferred from node 1, then d_1 = degree(1) = 5. If 7 handling units need to be transferred into node 3, then d_3 = degree(3) = 7.

Step 3: Solve the relaxed problem

Solve the Minimum Cost Network Flow Problem (MCNFP) which corresponds to formulation (4.17) – (4.22) without constraint set (4.21). The solution from the MCNFP provides the required arcs and non-required arcs so that $G = (N, A)$ is balanced.

Step 4: Check the connectivity of graph $G = (N, A)$

Check the connectivity of graph $G = (N,A)$ using a procedure from Rosen and Michaels (2000). Assume all nodes are in list K. Select some node x to place in L. Build a list of the nodes (L) which can be reached from x. Each time a new node is added to this list (L), the neighboring nodes are checked to see if they should be added. Finally the list is checked to see if the list covers the whole graph. If nodes remain in K, start from another node which is not in L and repeat the above again until all sub tours are found. The procedure is summarized in Figure 6.1.

```
Begin
  Choose a node x from N;
  Make a list L of nodes reachable from x, and another list K of nodes to be explored;
  L = K = {x};
  While K is nonempty, Do
      For each edge (y, z) where y ∈ K
          If z is not in L, add z to both L and K;
      End For
    If |L| < |N|
      Return disconnected;
  Else
      Return connected;
End
```

Figure 6.1 Connectivity Check Procedures

Step 5: Connect sub tours in the graph *G = (N, A)*

If disconnected sub tours are found, the Minimum Spanning Tree algorithm (Graham & Hell, 1985) is used to identify the set of edges (*M*) to connect the sub tours. For each spanning edge that connects the sub tours, make two copies of this edge, associating one direction with one edge and the opposite direction with the other edge.

Figure 6.2 Connect the Sub-tour

For example, assume two sub tours are identified as follows: (1, 2, 3, 4) and (5, 6, 7). Using Minimum Spanning Tree Algorithm, the edge (3, 5) is found to connect two sub tours as illustrated in Figure 6.2. Two copies of edge (3, 5) are made with opposite directions to connect the network and ensure the network is still balanced.

Step 6: Form a network that consists of required arcs and selected non-required arcs such that $G = (N, A \cup M)$

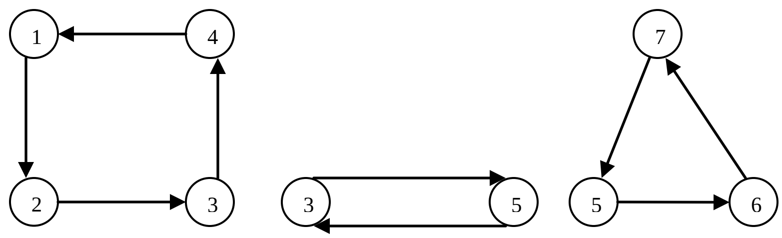
Required arcs, non required arcs and the arcs to connect sub tours form a network $G = (N, A \cup M)$. This network serves as the basis for the next step to find a sequence.

Step 7: Apply a Euler tour algorithm to find a sequence

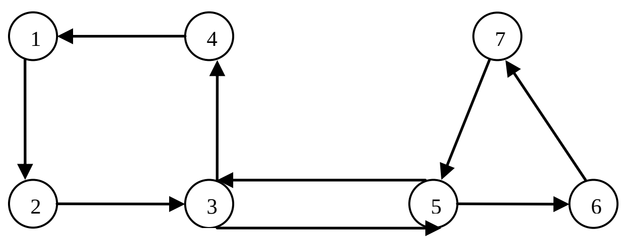
Several Euler tour algorithms (Christofides, 1975; Edmonds & Johnson, 1973; Rosen & Michaels, 2000) can be used to construct an Euler tour to find a sequence for the worker. In this research, an algorithm by Rosen and Michaels (2000) is used. This algorithm is based on the observation that if *C* is any cycle in an Euler graph, then after removing the edges of *C*, the remaining connected components will also be Euler graphs. The algorithm can be summarized in the two following steps and illustrated in Figure 6.3:

a. Find all cycles on the graph *G*;

b. Splice the cycles to form an Euler tour.



a. Find all cycles on the graph



b. Splice the cycles to form an Euler tour

Figure 6.3 Find the Euler Tour

Assume three cycles are found as follows: (1→2→3→4→1) (3→5→3) and (5→6→7→5). Using Euler tour construction algorithm, three cycles are spliced at node 3 and 5: (1→2→(3→(5→6→7→5)→3)→4→1). As illustrated in Figure 6.3, the Euler tour is 1→2→3→5→6→7→5→3→4→1.

To implement the Balance-and-Connect Algorithm for hub operations, a list of shipments (indicating the shipment number, handling units for this shipment, origin trailer and destination trailer for this shipment) for the worker is required. After the unloading and loading sequence is found from the algorithm, the sequence associated with each shipment is added to the shipment list. This shipment sequence list is used for the worker to unload and load shipments accordingly.

When applying Euler tour algorithms to find the sequence for each worker, the precedence of the shipment inside the origin trailer should be also followed. The precedence of the shipment inside the origin trailer is the order that the shipments are physically placed in the trailer. For example, if shipment 2 is physically placed in the trailer behind shipment 1, then in the shipment sequence, shipment 1 should be transferred before shipment 2 is transferred. An illustrative example for each of the steps of Balance-and-Connect Algorithm is provided in Appendix C.

The solution from the Balance-and-Connect Algorithm provides a feasible solution to the FSP for single worker as shown by Proposition 6.1.

Proposition 6.1 The Balance-and-Connect Algorithm for single worker produces an Euler tour for a worker to traverse all of the required arcs.

Proof. The solutions from step 2 with the arcs $\in R$ produce a set of cycles, each of which is consistent with arc directions. The in-degree and out-degree of each node are thus equal. The spanning tree edges created in step 4 connect the circles. Since two of each spanning edges are added, with one edge having the opposite direction from the other, the in-degree and out-degree of each vertex in G are still equal. Thus step 6 will produce an Euler tour from $A \cup M$ **.** □

Although the Balance-and-Connect Algorithm generates a feasible tour to transfer the shipments, the tour may not necessarily be optimal. In this research the Balance-and-Connect Algorithm is compared to the trailer-at-a-time and Nearest Neighbor algorithm.

6.1.2 Trailer-at-a-time for the FSP

Trailer-at-a-time is a widely used approach or rule for unloading and loading shipment operations in the LTL hub. With this approach, a worker transfers all shipments on an origin trailer before transferring shipments on another origin trailer. Each time a worker completes one shipment on a trailer, the worker returns to the same trailer for another shipment. After the worker finishes all shipments on that trailer, the worker moves to another origin trailer to transfer the shipments on that trailer. This process continues until all trailers assigned to the worker finish. Since the worker works on one trailer each time, this process is called trailer-at-a-time. The trailer-at-a-time approach is summarized in the following steps:

Step 1: Arbitrarily pick an origin trailer; Transfer the handling units for the first shipment on the trailer to the appropriate destination trailer;

Step 2: Return to the same origin trailer for transferring the handling units for next available shipment;

Step 3: Go to Step 2 until there are no shipments on the origin trailer;

Step 4: Go to Step 1 until there are no origin trailers left.

Trailer-at-a-time provides a practical solution approach to the FSP that is easily implemented in hub operations. The hub manager simply assigns trailers to workers and workers complete individual origin trailer without any pre-determined freight unloading and loading sequences. The disadvantage of this approach is that the travel distance of empty forklift movements might be unnecessarily long, compared to the travel distance with more efficient freight sequencing approaches, and thus result in long completion time to finish all shipments.

6.1.3 Nearest Neighbor Algorithm for the FSP

An alternative approach for solving the FSP is to use construction heuristics. Construction heuristics are used to construct a solution in a progressive manner. At each iteration, a partial solution is obtained and an extension of this solution is constructed by selecting one of a number of options available. A well-known construction heuristic is the Nearest Neighbor (NN) algorithm. The Nearest Neighbor algorithm was one of the first algorithms to address the traveling salesman problem (TSP). For the TSP, suppose that a partial tour has been constructed in previous iterations. This open tour has two ends, i.e., two cities that are currently linked only with a single city. One of these two cities is then linked to the city that is closest. This algorithm is myopic since it only considers the best possible next step. This approach quickly generates a tour, but not necessarily an optimal tour.

As described in Brown (2003), the Nearest Neighbor algorithm can be modified for finding the freight unloading and loading sequence for dock workers. A slight difference appears between NN for TSP and for this problem. For the NN algorithm for the TSP, the next closest node is added to the tour. For the NN algorithm for the FSP, however, a required arc is always chosen (rather than going to any nearest neighbor). At each destination trailer node, the worker goes to the next closest origin trailer node.

The Nearest Neighbor algorithm for the FSP is summarized in the following steps:

Step 1: Set up a network $G = (N, R)$ where N is the set of doors which contain the origin and destination trailers and R is the set of all required trips. Start with an arbitrary origin trailer.

Step 2: Transfer an arbitrary shipment on the manifest to the appropriate destination trailer.

Step 3: Find the origin trailer which has the shortest distance from the current trailer, go to that origin trailer and transfer an arbitrary shipment.

Step 4: Repeat Step 3 until no shipments remained on the manifest.

The Nearest Neighbor algorithm is easy to implement and executes quickly. Due to its myopic approach, however the Nearest Neighbor algorithm sometimes misses shorter routes which are easily noticed with human insight (Simchi-Levi & Bramel, 1998).

6.1.4 Case Study for the FSP for a Single Worker

Five data sets are chosen to compare the Balance-and-Connect Algorithm, the Nearest Neighbor algorithm, and the trailer-at-a-time approach for the FSP for a single worker. These data sets are the same data sets used in the trailer-to-door assignment problem in the Chapter 5. For each data set, the trailer-to-door assignment problem is solved using dynamic approach with simulated annealing. The characteristics of the data sets are summarized in Table 6.1.

Table 6.1 Data Sets for Case Study for the FSP

Data Set	Origin Trailers	Destination Trailers	Number of Shipments	Dock Doors	Number of Workers
1	16	15	178	31	1
2	16	15	178	31	1
3	16	16	201	32	1
4	17	17	173	34	1
5	62	33	652	95	1

The Trailer-at-a-Time approach and Nearest Neighbor are implemented in C++ and run on an Intel Pentium 4 2.8 GHz computer. To implement the Balance-and-Connect Algorithm, the model (4.17)–(4.22) without constraint (4.21) is written in AMPL 10.0 and solved with CPLEX 10.0 on an Intel Pentium 4 2.8GHz computer. The AMPL model and data for the five data sets are included in the Appendix B. The solution is then checked for connectivity, and the minimum spanning tree algorithm (Graham & Hell, 1985) is applied to connect the sub tours if necessary.

The results of total travel distance (feet) for the Trailer-at-a-Time (TAAT), Nearest Neighbor (NN) and the Balance-and-Connect Algorithm (BCA) are summarized in the Table 6.2. For these data sets, the Balance-and-Connect Algorithm outperforms the Trailer-at-a-Time and NN approach for total travel distance in all the cases. The percentage improvement of the BCA over the TAAT in terms of total travel distance varies from 10% to 27%. The percentage improvement of the BCA over the NN in terms of total travel distance also varies from 10% to 27%. These results demonstrate the substantial opportunity for reducing the total distance traveled by worker during the hub operations using Balance-and-Connect Algorithm.

As the problem size increases, the savings from the Balance-and-Connect Algorithm increased as well. As the number of handling units (required arcs) increase, more alternative routes exist for return trips for the worker. Thus the total savings from return trips in terms of total travel distance increased.

Table 6. 2 Total Travel Distance for Heuristics for Single Worker FSP

Data Set	TAAT (feet)	NN (feet)	BCA (feet)	BCA Improvement over TAAT	BCA Improvement over NN
1	100284	98679	89137	11%	10%
2	99954	101092	89726	10%	11%
3	79072	78067	68908	13%	12%
4	44892	46059	37117	17%	19%
5	477699	478571	349132	27%	27%

The results of total time (minutes) for the trailer-at-a-time (TAAT), Nearest Neighbor (NN) and the Balance-and-Connect Algorithm (BCA) are summarized in Table 6.3.

As shown, the BCA outperforms the TAAT and NN for total time as well. The percentage improvement of total time of the BCA over the TAAT and NN varies from 2% to 10%. The improvements on the total time from the BCA are not as significant as the improvements on the total travel distance from the BCA. The BCA improves hub operations through the routing optimization and the travel distances for the hub worker are minimized. The unloading and loading times are fixed costs, and improvements for the total completion time are limited by these fixed cost.

Table 6.3 Total Time for Heuristics for Single Worker FSP

Data Set	TAAT (min)	NN(min)	BCA (min)	BCA Improvement over TAAT	BCA Improvement over NN
1	2079	2040	1991	4%	2%
2	2064	2053	2001	3%	3%
3	1568	1558	1516	3%	3%
4	803	800	768	4%	4%
5	5766	5743	5201	10%	9%

6.2 Solutions Approaches for the FSP for *k* Workers

Multiple dock workers are often available to transfer freight during the hub operations. In this section, the Balance-and-Connect Algorithm and the Assign First Route Second Algorithm for multiple workers are discussed in detail.

6.2.1 Balance-and-Connect Algorithm for *k* Workers

The Balance-and-Connect Algorithm for FSP for *k* workers is developed based on formulation (4.23)-(4.30). In the formulation, constraint (4.29) assures that the network is connected. If this constraint is removed, the solution may include several disconnected sub-tours. The Balance-and-Connect Algorithm for *k* workers solves the relaxation model first and then checks for connectivity. If sub-tours exist, a minimum spanning tree algorithm can be used to generate a connected graph which contains a Euler tour. Finally, using a Euler tour finding procedure, a Euler tour is generated to correspond to the sequence for unloading and loading shipments for each worker. This sequential approach will not necessarily be optimal for the original problem because it can not address the interaction between the balancing and connecting steps. The Balance-and-Connect Algorithm for *k* workers is summarized as follows:

Step 1: Setup the directed RPP network.
Step 2: Determine the degree for each node.
Step 3: Solve the relaxed problem;
Step 4: Check the connectivity, and

Connect the sub tour using the minimum spanning tree algorithm if necessary.

Step 5: Form a network that consists of required arcs and selected non-required arcs.

Step 6: Apply Euler tour algorithm to find the sequence for each worker.

The steps are similar to the Balance-and-Connect Algorithm for single worker. The primary difference is that the relaxed model solved in Step 2 is based on the IP model (4.23) – (4.30) for FSP for *k* workers.

The IP model (4.23) – (4.30) for the FSP for *k* workers is modeled in AMPL without connectivity constraint (4.28) and solved using CPLEX solver. After obtaining near optimal solutions (by setting gap limitation or time limitation for the solver), solutions for each worker are checked for connectivity. The minimum spanning tree algorithm is applied to connect the sub-tour if necessary. Then the Euler tour algorithm is used to find the freight sequence for each worker. The Balance-and-Connect Algorithm for *k* workers is illustrated in Figure 6.4.

The Balance-and-Connect Algorithm for *k* workers generates a feasible solution for each worker in the hub operations as shown in the following proposition.

Proposition 6.2 The Balance-and-Connect Algorithm for *k* workers produces *k* Euler tours for *k* workers to traverse all of the required arcs.

Proof. The solutions from step 3 produce a set of cycles, each of which is consistent with arc directions. The in-degree and out-degree of each node are thus equal. The spanning tree edges created in step 4 connect the circles for each worker. Since two of each spanning edge are added, with one edge having the opposite direction from the other, the in-degree and out-degree of each node are still equal. Thus step 6 will produce an Euler tour for each worker. □

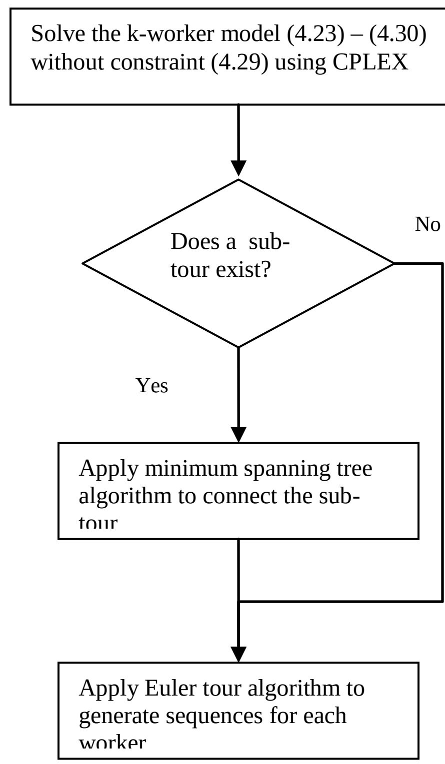


Figure 6.4 Balance-and-Connect Algorithm for FSP for *k* workers

The Balance-and-Connect Algorithm generates a feasible tour to transfer the shipments, although not necessarily an optimal tour. Several industrial data sets will be used to test the effectiveness of the Balance-and-Connect Algorithm and compare the results to the Assign First Route Second Algorithm for *k* workers.

6.2.3 Assign First and Route Second Algorithm for *k* workers

In actual hub operations, a dock supervisor typically assigns trailers to dock workers mostly based on judgment and experience. The goal is to assign trailers to dock workers such that each dock worker has fairly equal amounts of shipments to transfer so the completion time of all transfers is minimized. After workers are assigned trailers, the trailer-at-a-time sequence rule is typically used to perform the shipment transferring operations.

Another approach to this assignment problem is the Balanced Trailer-at-a-Time (BTAAT) approach described by Brown (2003). The basic idea of this approach is to use a greedy construction heuristic that assigns the available trailer with the largest estimated transfer time to the dock worker with the smallest workload assigned. To estimate transfer time, the trailer-at-a-time approach is assumed. After workers are assigned trailers, the trailer-at-a-time sequence rule is used to perform the shipment transferring operations. The Balanced Trailer-at-a-Time procedure is summarized in the following steps:

Step 1: Calculate workload for each origin trailer as described by Brown (2003).

Step 2: Sort origin trailers in non-increasing order of workload. Sort workers in non-decreasing order on workload;

Step 3: Assign the origin trailer with the largest workload to the worker with the least workload;

Step 4: Repeat step 4 until there is no origin trailer left.

Step 5: For each worker perform trailer-at-a-time to transfer all shipments.

An illustration of Balanced Trailer-at-a-Time assignment is shown in Figure 6.5.

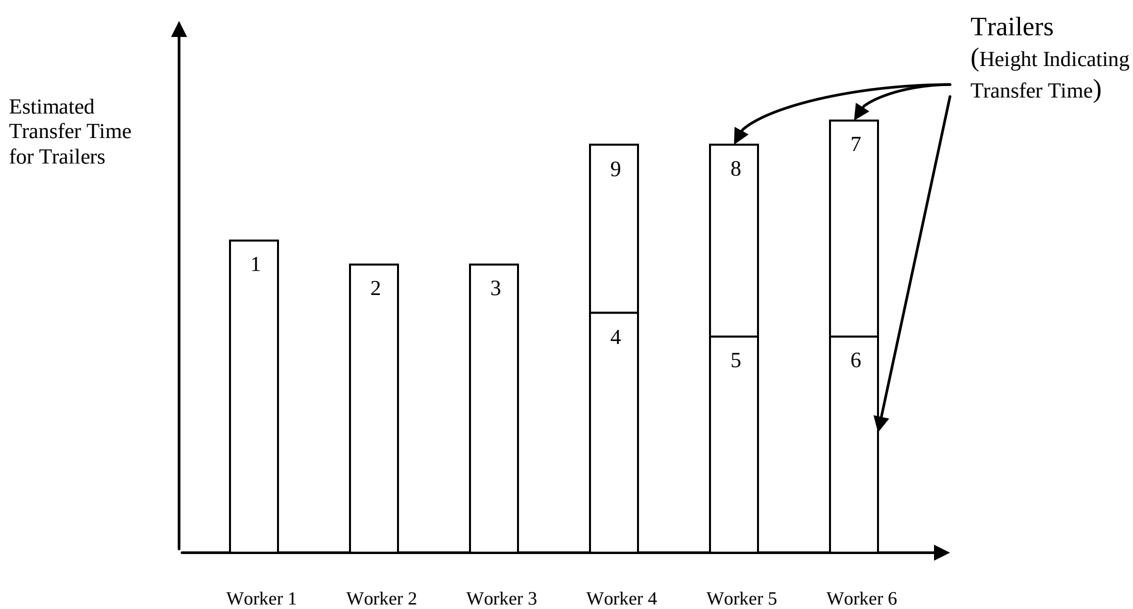


Figure 6.5 Balanced Assignments for Workers

In this example, nine origin trailers are available to assign to six workers. The estimated transfer for each trailer is calculated (using a trailer-at-a-time approach). The nine origin trailers are then sorted in non-increasing order of transfer. The trailer with the largest workload (trailer 1) is assigned to worker one. The second longest travel time trailer (trailer 2) is assigned to worker two and so on. After the first six trailers are assigned, trailer seven is assigned to worker 6 since worker 6 is the worker currently with least workload. The remaining trailers are assigned to the workers accordingly until no trailers remain.

The Assign First and Route Second Algorithm uses the Balanced Trailer-at-a-Time assignment approach from Brown (2003) to generate a relatively balanced trailer assignment for each worker. After the trailers are assigned to dock workers, the Balance-and-Connect Algorithm for a single worker is performed for each worker to find the best freight unloading and loading sequence. The Balance-and-Connect Algorithm for a single worker is based on IP formulation (4.17)-(4.21). The Balance-and-Connect Algorithm for a single worker is to solve model (4.17)-(4.21) without constraint (4.20). Since the connectivity constraint (4.20) is dropped, the solution might include sub-tours. A minimum spanning tree algorithm (i.e., Graham & Hell, 1985) is used to connect the graph if necessary and finally a Euler tour algorithm (Rosen and Michaels, 2000) is used to find the freight sequence for the worker. Hence for the hub worker the total travel distance is minimized, and the make-span for transferring all shipments is also minimized. The AFRSA is summarized in the following steps:

Step 1: Use Balanced Trailer-at-a-Time approach from Brown (2003) to assign trailers to the workers.

Step 2: Setup the directed RPP network for each worker *k*;

Step 3: Solve model (4.17) – (4.21) without (4.20) for each worker *k*;

Step 4: Check the connectivity for each worker *k***;** Apply minimum spanning tree algorithm to connect the sub-tours if necessary.

Step 5: Apply a Euler tour algorithm to find a sequence for each worker *k*.

In the following section, the AFRSA is tested using an industrial data set and compared to the BCA for the FSP for *k* workers.

6.2.4 Illustrative Example for the FSP for *k* Workers

A data set for one night of hub operations is evaluated using Balanced Trailer-at-a-Time (BTAAT), Assign First and Route Second Algorithm (AFRSA), and the Balance-and-Connect Algorithm (BCA). BTAAT us described by Brown (2003). The AFRSA and the BCA are developed in Section 6.2.1 and 6.2.2. The data set includes 201 shipments for transfer by 6 workers at a hub with 32 doors.

For the Balance-and-Connect Algorithm, model (4.23) – (4.30) without constraint (4.29) is formulated in AMPL 10.0 and solved with CPLEX10.0 (by setting the gap level as 0.01%). The AMPL model and data are included in the Appendix B. The connectivity is checked and the Euler tour algorithm is applied to find the sequence for each worker.

For the Assign-First-Route-Second Algorithm, the balanced trailer-at-a-time assignment procedure is applied first for 6 workers. Table 6.4 shows the assignment for the 6 worker.

Table 6.4 Trailer Assignments to Worker 1-6

Worker	Number of Trailers	Number of Shipments	Number of HU
1	3	33	125
2	4	37	167
3	4	30	205
4	1	36	120
5	1	24	112
6	3	41	106

After generating the directed RPP network $G_k = (N_k,\ A_k' \cup R_k)$ for each k (k = 1, 2, 3, 4, 5, 6), the model (4.17) – (4.21) without (4.20) is written in AMPL 10.0 and solved with CPLEX10.0. The connectivity is checked and the Euler tour algorithm is applied to find the sequence for each worker.

The travel distance for each worker and total travel distance using three algorithms are summarized in Table 6.5. As shown, the BCA has the least total travel distance (69040 feet) for

6 workers, compared to the total travel distance for the BTAAT (79072 feet) and the AFRSA (74465 feet).

Table 6.5 Travel Distance (feet) for the FSP with 6 workers

Worker	Balanced trailer-at-a-time	Assign First Route Second Algorithm	Balance-and-Connect Algorithm
1	13646	11780	10563
2	14928	14103	9826
3	16068	15876	12583
4	11200	11200	13947
5	9540	9540	11567
6	13690	11966	10554
Sum	**79072**	**74465**	**69040**

The transfer time for each worker and total transfer time using three algorithms are summarized in Table 6.6. As shown, the transfer time for the bottleneck worker for the BCA is the least (252.59 minutes), compared to the transfer time for the bottleneck worker for the BTAAT (369.53 minutes) and the AFRSA (367.73 minutes). Since the make-span time for hub operations is determined by the transfer time for the bottleneck worker, the BCA substantially reduces the make-span time to complete all shipments. The BCA reduced the make-span time 116.68 minutes from the BTAAT and 114.88 minutes from the AFRSA. Since the BCA allows the workers to share origin trailers, the workload for each worker is distributed more evenly (Figure 6.6). The BCA also has the least total transfer time (1516.69 minutes), compared to the total transfer time for the BTAAT (1568.33 minutes) and the AFRSA (1540.08 minutes).

Table 6.6 Transfer Time (minutes) for the FSP with 6 workers

Worker No.	Balanced trailer-at-a-time	Assign First Route Second Algorithm	Balance-and-Connect Algorithm
1	253.02	233.28	252.85
2	305.58	304.61	252.59
3	369.53	367.73	252.80
4	223.48	223.48	252.84
5	204.64	204.64	252.80
6	212.08	206.34	252.81
Sum	**1568.33**	**1540.08**	**1516.69**

Figure 6.6 Comparison of Completion Time for 6 Workers in 3 Algorithms

Table 6.7 summarizes the improvement of the BCA and the AFRSA over the BTAAT for the make-span time, total transfer time and total distance. As shown, the BCA reduces the make-span time by 32% over BTAAT and by 31% over AFRSA. The BCA reduces the total transfer time by 3% over BTAAT and by 2% over AFRSA. The BCA also reduces total travel distance by 13% over BTAAT and 7% over AFRSA.

The AFRSA also reduces the make-span time, the total transfer time and the total distance compared to the BTAAT. For each worker with multiple trailers assigned, the travel distance and the transfer time from the AFRSA is also less than the BTAAT. For the workers with only one trailer assigned, the travel distance and the completion time from the AFRSA are remained the same as the BTAAT. Overall, when comparing the AFRSA to BTAAT, the make span is reduced by 1%, the total completion time is reduced by 2%, and the total distance is reduced by 6%. This illustrates the potential improvement opportunity when multiple origin trailers are assigned to a worker and efficient return trips are planned.

Table 6.7 Improvements from the BCA and the AFRSA

Measurements	AFRSA Improvement to BTAAT	BCA Improvement to BTAAT	BCA Improvement to AFRSA
Make-span time	1%	32%	31%
Total transfer time	2%	3%	2%
Total travel distance	6%	13%	7%

The number of trailers assigned for each worker is shown in Table 6.8. As shown, the number of trailers assigned to each worker with the BCA is more than the number of trailers assigned to each worker for the BTAAT and the AFRSA. Solutions from Balance-and-Connect Algorithm provide more balanced workloads for each worker and shorter total distance which results in shorter processing time for hub operations but more sharing of origin trailers for workers.

Table 6. 8 Number of Trailers Assigned for the FSP for 6 workers

Worker No.	Balanced trailer-at-a-time	Assign First Route Second Algorithm	Balance-and-Connect Algorithm
1	3	3	13
2	4	4	11
3	4	4	11
4	1	1	12
5	1	1	6
6	3	3	12

6.2.5 Case Study for the Freight Sequencing Problem for *k* Workers

The effectiveness of the following solution approaches for the FSP with *k* workers are evaluated:

- Balanced trailer-at-a-time (BTAAT),
- Balanced trailer-at-a-time with offloading (BTAATWO),
- Nearest Neighbor algorithm (NN),
- Assign First Route Second Algorithm (AFRSA), and
- Balance-and-Connect (BCA).

The BTAAT, BTAATWO and NN are from Brown (2003). The descriptions of BTAATWO and NN are included in Appendix F. Five data sets from two LTL carriers are used for the evaluation. The data sets, summarized in Table 6.9, are the same data sets used in Section 6.14

except multiple workers are available. The results for the following performance measures are analyzed:

- Bottleneck distance
- Bottleneck time
- Total travel distance
- Total time

Table 6.9 Data Sets Summary for *k* Workers

Data Set	Origin Trailers	Destination Trailers	Number of Shipments	Dock Doors	Number of Workers
1	16	15	178	31	6
2	16	15	178	31	6
3	16	16	201	32	6
4	17	17	173	34	3
5	62	33	652	95	10

The bottleneck distance results are summarized in Table 6.10. As shown, the least bottleneck distance occurs with the BCA for all data sets. The BTAATWO and NN have much less bottleneck distance, compared to the BTAAT and the AFRSA. Further investigation of the data shows that three data sets (data sets 1, 2 and 5) have a large trailer (in terms of transfer time) that dominates the bottleneck distance. Since the BTAATWO, NN, and the BCA allow sharing of trailers, these algorithms tend to perform well in these cases. Table 6.11 summarized the percentage improvement in bottleneck distance of the four algorithms over BTAAT. As shown, the BCA reduces the bottleneck distance over the BTAAT by 34% to 77%. The reduction of the bottleneck distance for BTAATWO and NN varies from 10% to 60% over BTAAT.

Table 6.10 Bottleneck Distance for the FSP for *k* Workers

Data Set	BTAAT (feet)	BTAATWO (feet)	NN (feet)	AFRSA (feet)	BCA (feet)
1	38821	18562	18878	38821	14704
2	39261	18387	20715	39261	14438
3	16068	14446	11445	15876	10563
4	17610	15566	15182	14524	11700
5	135677	70751	54020	135677	31756

Table 6.11 Improvements for the Bottleneck Distance Compared to the BTAAT

Data Set	BTAATWO	NN	AFRSA	BCA
1	52%	51%	0%	62%
2	53%	47%	0%	63%
3	10%	29%	1%	34%
4	12%	14%	18%	34%
5	48%	60%	0%	77%

The bottleneck time results are summarized in the Table 6.12. As shown, the BCA has the least bottleneck time for all data sets. The BTAATWO and NN also have much less bottleneck time, compared to the BTAAT and the AFRSA. Again, since the BTAATWO, NN and the BCA allow sharing of trailers, these algorithms reduced bottleneck time substantially where there exits dominate trailers in the hub operations. Table 6.13 summarized the percentage improvement in bottleneck time of the four algorithms over BTAAT. As shown, the BCA reduces bottleneck time over the BTAAT by 17% to 68%. The reduction of the bottleneck time for BTAATWO and NN varies from 6% to 63% over BTAAT..

Table 6.12 Bottleneck Time for the FSP for *k* Workers

Data Set	BTAAT (min)	BTAATWO (min)	NN (min)	AFRSA (min)	BCA (min)
1	921	345	343	922	332
2	926	361	371	924	334
3	369	347	302	367	253
4	308	277	277	297	256
5	1624	729	929	1624	522

Table 6.13 Improvements for the Bottleneck Time Compared to the BTAAT

Data Set	BTAATWO	NN	AFRSA	BCA
1	63%	63%	0%	64%
2	61%	60%	0%	64%
3	6%	18%	1%	31%
4	10%	10%	4%	17%
5	55%	43%	0%	68%

The total distance for each algorithm is summarized in the Table 6.14. As shown, AFRSA and BCA reduce the total distance over BTAAT for all data set. The BCA generates the least total travel distance for all data sets. With the BCA, the empty forklift travel distances (non-required arcs) are reduced, compared to all other algorithms. The percentage improvement in total distance of the four algorithms over the BTAAT is summarized in Table 6.15. The reduction in total distance by the BCA varies from 10% to 27% over BTAAT. Using the BCA to decrease the total distance traveled during the hub operations offers LTL carriers the potential benefits of decreasing the cost (labor cost, fuel cost) of inefficient the hub operations.

Table 6.14 Total Travel Distance for the FSP for *k* Workers

Data Set	BTAAT (feet)	BTAATWO (feet)	NN (feet)	AFRSA (feet)	BCA (feet)
1	100284	100820	98015	97808	89159
2	99954	101449	99821	97029	89770
3	79072	80103	77821	74465	69040
4	44892	44890	44791	40010	37117
5	477699	478548	472312	420787	350615

Table 6.15 Improvements for the Total Distance Compared to the BTAAT

Data Set	BTAATWO	NN	AFRSA	BCA
1	1%	2%	2%	11%
2	1%	0%	3%	10%
3	1%	2%	6%	13%
4	0%	0%	11%	17%
5	0%	1%	12%	27%

The total time for each algorithm is summarized in the Table 6.16. As shown, AFRSA and BCA reduce the total time over BTAAT for all data sets. The BCA generates the least total time for all data sets. The percentage improvement in total time of four algorithms over the BTAAT is summarized in Table 6.17. This result illustrates the fact that the BCA is improving the hub operations through the routing optimization thus the travel distances and travel time for both bottleneck worker and other workers are minimized. The unloading and loading time are considered as fixed costs and improvements for the total completion time are limited by these fixed cost. In the future LTL carriers have the opportunity to improve the efficiency of loading and unloading process in the hub operations.

The BCA performs best for total distance and total time for all data sets. The NN and the AFRSA performs at least as well as the BTAAT in all data sets in terms of total distance and total time. The improvement of the NN and the AFRSA over the BTAAT for total distance and total time is less than the improvement of the BCA. The AFRSA may be easier to implement at the hub, compared to the BCA, since there is no sharing of origin trailers. The total time, total distances and the bottleneck time (in some cases), however, can be improved through the routing optimization process for each worker.

Table 6.16 Total Time for the FSP for *k* Workers

Data Set	BTAAT (min)	BTAATWO (min)	NN (min)	AFRSA (min)	BCA (min)
1	2079	2052	2040	2035	1991
2	2064	2055	2053	2031	2001
3	1568	1568	1558	1539	1517
4	803	804	800	781	769
5	5766	5770	5743	5506	5206

Table 6.17 Improvements for the Total Time Compared to the BTAAT

Data Set	BTAATWO	NN	AFRSA	BCA
1	1%	2%	2%	4%
2	0%	1%	2%	3%
3	0%	1%	2%	3%
4	0%	0%	3%	4%
5	0%	0%	5%	10%

6.3 Summary of Results

In this chapter, solution approaches for the freight sequencing problem (FSP) for single worker and the FSP for *k* workers are discussed. The Balance-and-Connect Algorithm (BCA) for a single worker, BCA for *k* workers, and the Assign First and Route Second Algorithm (AFRSA) for k workers are developed in the chapter. These algorithms are compared to the balanced trailer-at-a-time (BTAAT), the balanced trailer-at-a-time with offloading (BTAATWO) and the Nearest Neighbor algorithm (NN) using several industry data sets. The results demonstrate the substantial improvement in terms of completion time and travel distance for hub workers using the BCA and AFRSA to solve the FSP.

The results also highlight opportunities for future research. The solutions from the Balance-and-Connect Algorithm for *k* workers might result in different workers sharing the same origin trailer. While sharing of origin trailers offers the increased opportunity for balanced workload across each worker, the solution might result in blocking when multiple workers try to unload the same origin trailer at the same time. To restrict the number of workers that can be assigned to an origin trailer, additional constraints could be added to the model for FSP for *k* workers and the Balance-and-Connect Algorithm could be amended accordingly (as described in Appendix). Also, in some cases, the unloading process is constrained by precedence due to the order that the shipments are physically placed in the trailer. For example, if shipment 2 is physically placed in the trailer behind shipment 1, then in the shipment sequence, shipment 1 should be transferred before shipment 2 is transferred since it is not practical for the worker to take the shipment 2 out of trailer before he transfer shipment 1. For a single worker, the precedence constraints are ensured for all the algorithms. For multiple workers, with BTAAT, NN, and AFRSA, the precedence constraints are easily ensured. With BTAATWO and BCA, however, the precedence constraints may not be ensured for multiple workers without adjustment. This provides opportunities for future research on the FSP.

Chapter VII

Simulation Study

Simulation is often used to describe and measure the performance of a system when one or more values of the independent variables are dynamic or uncertain. In the trailer-to-door assignment problem and the freight sequencing problem addressed in previous chapter, the following important static and deterministic assumptions were made:

- All origin and destination trailers are available at the beginning of the operation and are available throughout the hub operations;
- Dock doors are available for all trailers;
- No congestion or blocking occurs in the hub operations;
- The travel speed of the forklifts and the unload and load times are constant.

In actual hub operations, however, trailers may arrive at the hub at different times. For example, some trailers are not available at the beginning of the hub operations. All the hub workers may also not be available at the beginning of the operations. The hub may have insufficient dock doors to accommodate all the origin and destination trailers. Blocking might occur when multiple workers try to unload and load the same trailer. The travel speeds and unload and load times likely vary.

To analyze dynamic and stochastic characteristics of hub operations, a discrete event simulation model is developed to evaluate the trailer-to-door assignment and the freight unloading and loading sequence approaches with the following effects:

- Potential blocking at the origin and destination trailers; and
- Workers travel speed and unload and load time as random variables.

The simulation model also provides an estimation of performance measures which are important from an operational perspective. In this chapter, a discrete simulation model is developed to simulate existing hub operations process, evaluate the impact of solutions from previous chapters in dynamic and stochastic settings, and provide more accurate estimation of performance measures of hub operations. The following sections describe performance measures, simulation experiments and simulation experiment results.

7.1 Performance Measures

The main performance measures evaluated by the simulation model include:

- Total worker time
- Total travel distance
- Transfer time
- Total worker travel time
- Bottleneck travel time
- Average workload balance ratio

Total worker time is the sum of the time for each of the workers to transfer the shipments. *Total travel distance* is the sum of the distances that each of the workers travel to transfer all shipments during the hub operations. The *transfer time* is the time span from the start of the hub operations to the time the last shipment is completed. The transfer time is also referred to as the make-span of the hub operations. *Total worker travel time* is the sum of the travel times of all the workers. *Bottleneck travel time* is the travel time for the bottleneck worker. Finally, the *average workload balance ratio* reflects the balance in terms of work time among all workers. Assume T_k is the time required for worker k to complete transferring shipments. Then $\max_k\{T_k\}$ is the time for the bottleneck worker to complete shipment transfers. The *average workload balance* ratio is defined as follows:

$$Average\ workload\ balance\ ratio = \frac{1}{K}\sum_{k=1}^{K}\frac{t_k}{\max\limits_{k}\{T_k\}}$$

For example, assume a bottleneck worker requires100 minutes and another worker requires 90 minutes. The workload balance ratio for the bottleneck worker is 100%, the workload balance ratio for the other worker is 90%, and the average workload balance ratio among these two workers is 95%. This ratio reflects the balance in terms of work time among all workers. In general, a higher ratio indicates more evenly balanced workloads. These performance measures are more formally defined in Appendix E.

7.2 Simulation Experiments

Two main experiments are performed to evaluate the trailer-to-door-assignment and freight sequencing approaches. As described in Table 7.2, Experiment 1 analyzes the performance of the freight sequencing approaches for a single worker. Experiment 2 analyzes the performance of the hub layout and freight sequencing approaches for multiple workers. The abbreviations for the approaches are summarized in Table 7.1.

Table 7.1 Door Assignment and Freight Sequencing Approaches in Simulation Model

Trailer-to-Door Assignment Approach	Abbreviation
Semi-permanent assignment with Pair-wise Exchange	SPW
Dynamic assignment with Pair-wise Exchange	DPW
Dynamic assignment with Genetic Algorithm	DGA
Dynamic assignment with Simulation Annealing	DSA
Freight Sequencing Approach	Abbreviation
Balanced Trailer-at-a-Time	BTAAT
Nearest Neighbor	NN
Trailer-at-a-Time with Offloading	TAATWO
Assign-First-Route-Second Algorithm	AFRSA
Balance-and-Connect Algorithm	BCA

Table 7.2 Simulation Experiments

Experiment No.	Purpose of Experiment	Configurations	Door Assignment	Freight Sequencing
1	Evaluate solution approaches for door assignment and freight sequencing with static and stochastic settings	Number of doors > Number of trailers Trailer Arrival Time: Static Worker Travel Speed: Stochastic Number of Workers: Single	1. DSA	1. BTAAT 2. NN 3. BCA
2	Evaluate solution approaches for door assignment and freight sequencing with static and stochastic settings	Number of doors > Number of trailers Trailer Arrival Time: Static Worker Travel Speed: Stochastic Number of Workers: Multiple	1.SPW 2.DPW 3.DGA 4. DSA	1. BTAAT 2. TAATWO 3. NN 4. AFRSA 5. BCA

Three data sets from LTL hub operations are used to perform the simulation experiments discussed. The characteristics of the data are summarized in the Table 7.3.

Table 7. 3 Data Set Characteristics for Simulation Experiment

Data Set	Origin Trailers	Destination Trailers	Number of Shipments	Number of Handling Units	Dock Doors	Number of Workers
1	16	16	201	835	32	6
2	16	15	178	1106	31	6
3	16	15	178	1110	31	6

The solution approaches in Chapter 5 and 6 are used to obtain the door assignment and freight sequence for each data set. The resulting solutions are then read into the simulation model. Simulation runs are then performed with the door assignment and freight sequencing combinations. The results from the mathematical models are also used to compare to the results from the simulation model to provide some verification for the simulation study (further described in Appendix E).

7.3 Experiment 1 Results and Analysis

The purpose of Experiment 1 is to evaluate the solutions for the freight sequencing problem with a single worker in the stochastic environment. For this experiment, the trailer-at-door solutions are obtained using a dynamic approach with simulated annealing. Each simulation run has 20 replications. Since only one worker performs hub operations, only three freight sequencing approaches are applicable: Balanced Trailer-at-a-time, Nearest Neighbor, and Balance-and-Connect. Thus three simulation runs are conducted for each data set.

The performance measures from the simulation runs for the three data sets are summarized in Table 7.4, Table 7.5, and Table 7.6, respectively.

Table 7.4 Performance Measure Results for Experiment 1 for Data Set 1

Simulation Run	Door Assignment	Freight Sequencing	Total Travel Distance (ft.)	Average Total Worker Travel Time (min)	Average Transfer Time (min)
1	DSA	BTAAT	80245	421	1643
2	DSA	NN	76851	389	1602
3	DSA	BCA	69102	362	1584

Table 7. 5 Performance Measure Results for Experiment 1 for Data Set 2

Simulation Run	Door Assignment	Freight Sequencing	Total Travel Distance (ft.)	Average Total Worker Travel Time (min)	Average Transfer Time (min)
1	DSA	BTAAT	99615	522	2144
2	DSA	NN	99455	520	2110
3	DSA	BCA	91470	480	2099

Table 7. 6 Performance Measure Results for Experiment 1 for Data Set 3

Simulation Run	Door Assignment	Freight Sequencing	Total Travel Distance (ft.)	Average Total Worker Travel Time (min)	Average Transfer Time (min)
1	DSA	BTAAT	100886	529	2156
2	DSA	NN	99120	521	2130
3	DSA	BCA	92796	487	2114

Across all data sets, NN reduces total travel distance, total worker travel time and transfer time for all data sets compared to BTAAT. Across all data sets, BCA out performs both NN and BTAAT with respect to total travel distance, total travel time and transfer time. BCA substantially reduces total distance (varying from 8% to 14%) and total worker travel time (varying from 8% to 14%), compared to BTAAT.

The percentage reduction by BCA on transfer time varies from 2% to 4% over BTAAT. These reductions are not as substantial as the reductions on total distance. The workload balancing capabilities of BCA are not evident when only one worker is in the hub operations.

Experiment 1 demonstrates that even with one worker in hub operations, adopting more efficient freight sequencing rules (i.e., BCA and NN) can reduce the total distance traveled during the hub operations.

7.4 Experiment 2 Results and Analysis

The purpose of Experiment 2 is to evaluate the solutions for the trailer-to-door assignment problem and freight sequencing problem for multiple workers in a dynamic and stochastic environment. Specifically, four trailer-to-door assignment approaches (semi-permanent assignment with Pair-wise Exchange, dynamic assignment with Pair-wise Exchange, dynamic assignment with Genetic Algorithm, and dynamic assignment with Simulated Annealing) and five freight sequencing approaches (Balanced Trailer-at-a-time, Trailer-at-a-time with Offloading, Nearest Neighbor, Assign-First-Route-Second, and Balance-and-Connect Algorithm) are used in Experiment 2. Thus 20 simulation runs are conducted and each simulation run has 20 replications. The average performance measures from each simulation run for each of the three data set one are collected and summarized in Table 7.7, Table 7.8, and Table 7.9. The variances of performance measures are summarized in Appendix E.

Table 7.7 Performance Measure Results Experiment 2 for Data Set 1

Simulation Run	Door Assignment	Freight Sequencing	Total Travel Distance (ft.)	Average Total Worker Travel Time (min)	Average Total Worker Time (min)	Average Worker Time (min)	Average Transfer Time (min)	Average Bottleneck Travel Time (min)	Average Workload Balance Ratio %
1	SPW	BTAAT	107175	562	1782	297	401	101	74%
2	SPW	NN	104590	548	1769	295	398	98	81%
3	SPW	TAATWO	106701	551	1770	295	389	101	82%
4	SPW	AFRSA	106551	559	1779	297	401	101	74%
5	SPW	BCA	100261	526	1746	291	305	108	95%
6	DPW	BTAAT	81423	427	1646	274	390	91	70%
7	DPW	NN	78932	413	1634	272	376	86	81%
8	DPW	TAATWO	81540	420	1641	274	371	87	85%
9	DPW	AFRSA	80819	424	1643	274	389	90	70%
10	DPW	BCA	76007	398	1618	270	280	87	96%
11	DGA	BTAAT	81004	420	1640	273	390	90	70%
12	DGA	NN	79504	417	1637	273	375	85	81%
13	DGA	TAATWO	80956	416	1634	272	370	87	84%
14	DGA	AFRSA	80093	420	1638	273	387	87	71%
15	DGA	BCA	73426	385	1605	268	274	72	98%

16	DSA	BTAAT	79958	419	1639	273	389	89	70%
17	DSA	NN	77458	406	1627	271	355	85	82%
18	DSA	TAATWO	79905	410	1630	272	367	87	85%
19	DSA	AFRSA	77959	409	1628	271	386	87	70%
20	DSA	BCA	70478	369	1589	265	269	68	98%

Table 7.8 Performance Measure Results Experiment 2 for Data Set 2

Simulation Run	Door Assignment	Freight Sequencing	Total Travel Distance (ft.)	Average Total Worker Travel Time (min)	Average Total Worker Time (min)	Average Worker Time (min)	Average Transfer Time (min)	Average Bottleneck Travel Time (min)	Average Workload Balance Ratio %
1	SPW	BTAAT	161734	848	2465	411	1152	395	36%
2	SPW	NN	160373	840	2459	410	462	240	74%
3	SPW	TAATWO	161856	842	2433	406	488	236	78%
4	SPW	AFRSA	161984	849	2465	411	1152	395	36%
5	SPW	BCA	160412	841	2355	393	449	183	87%
6	DPW	BTAAT	107218	562	2180	363	993	236	37%
7	DPW	NN	105929	555	2172	362	416	136	73%
8	DPW	TAATWO	107008	570	2168	361	421	140	77%
9	DPW	AFRSA	106725	559	2176	363	993	236	37%
10	DPW	BCA	103151	541	2136	356	390	115	91%
11	DGA	BTAAT	101134	530	2146	358	964	207	37%
12	DGA	NN	103480	543	2161	360	414	116	76%
13	DGA	TAATWO	101300	540	2158	360	420	120	80%
14	DGA	AFRSA	103473	542	2161	360	974	216	37%
15	DGA	BCA	101151	541	2158	360	386	115	93%
16	DSA	BTAAT	100804	528	2143	357	963	207	37%
17	DSA	NN	99806	523	2140	357	380	100	78%
18	DSA	TAATWO	100901	528	2140	357	390	102	79%
19	DSA	AFRSA	99557	522	2140	357	964	207	37%
20	DSA	BCA	92037	483	2098	350	367	94	95%

Table 7.9 Performance Measure Results Experiment 2 for Data Set 3

Simulation Run	Door Assignment	Freight Sequencing	Total Travel Distance (ft.)	Average Total Worker Travel Time (min)	Average Total Worker Time (min)	Average Worker Time (min)	Average Transfer Time (min)	Average Bottleneck Travel Time (min)	Average Workload Balance Ratio %
1	SPW	BTAAT	164753	859	2510	418	1171	403	37%
2	SPW	NN	160546	841	2433	406	434	208	74%
3	SPW	TAATWO	161940	843	2465	411	446	210	79%
4	SPW	AFRSA	163020	850	2502	417	1160	401	39%
5	SPW	BCA	161798	848	2473	412	451	184	88%

6	DPW	BTAAT	109115	580	2193	366	997	231	38%
7	DPW	NN	107867	564	2172	362	420	133	74%
8	DPW	TAATWO	109001	572	2170	362	420	138	81%
9	DPW	AFRSA	105939	555	2180	363	990	233	39%
10	DPW	BCA	104187	546	2169	362	386	113	94%
11	DGA	BTAAT	101245	529	2149	358	962	210	34%
12	DGA	NN	102989	541	2154	359	418	114	78%
13	DGA	TAATWO	101298	539	2155	359	420	119	79%
14	DGA	AFRSA	102983	540	2169	362	972	211	38%
15	DGA	BCA	101104	551	2175	363	387	116	94%
16	DSA	BTAAT	100920	530	2151	359	962	203	38%
17	DSA	NN	99087	520	2140	357	382	102	79%
18	DSA	TAATWO	100204	527	2120	353	388	100	78%
19	DSA	AFRSA	99008	518	2138	356	960	204	38%
20	DSA	BCA	93363	490	2111	352	367	94	96%

The impact of door assignment on the performance measures of hub operations is observed by comparing total distance, total worker time, transfer time, and average workload ratio from the dynamic approaches (DPW, DGA and DSA) to the same performance measures from the semi-permanent approach (SPW). The following paragraphs describe the results for data set one for total distance, total work time, transfer time, and average workload ratio. The results for data set 2 and 3 are provided in Appendix E.

Figure 7.1 illustrates the comparison of total distance for freight sequencing with each door assignment for data set 1. As shown, the total distance was reduced from more than 100,000 feet with the semi-permanent approach to less than 80,000 feet with the dynamic approaches. The improvements in total distance using dynamic door assignment approaches varied from 24% to 30%.

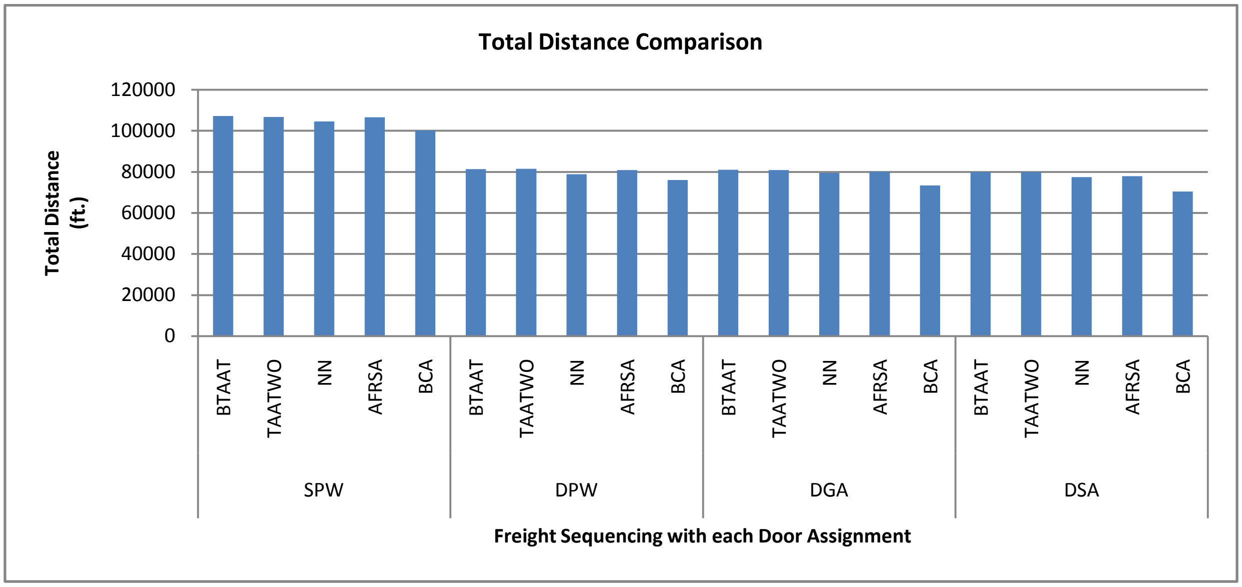


Figure 7.1 Total Distance Comparison from Data Set 1 for Simulation Experiment 2

Figure 7.2 illustrates the comparison of total worker time for freight sequencing with each door assignment from data set 1. As shown, the dynamic layouts reduce total worker time from more than 1750 minutes to less than1650 minutes. The improvements in total worker time with dynamic layouts varied from 7% to 9%.

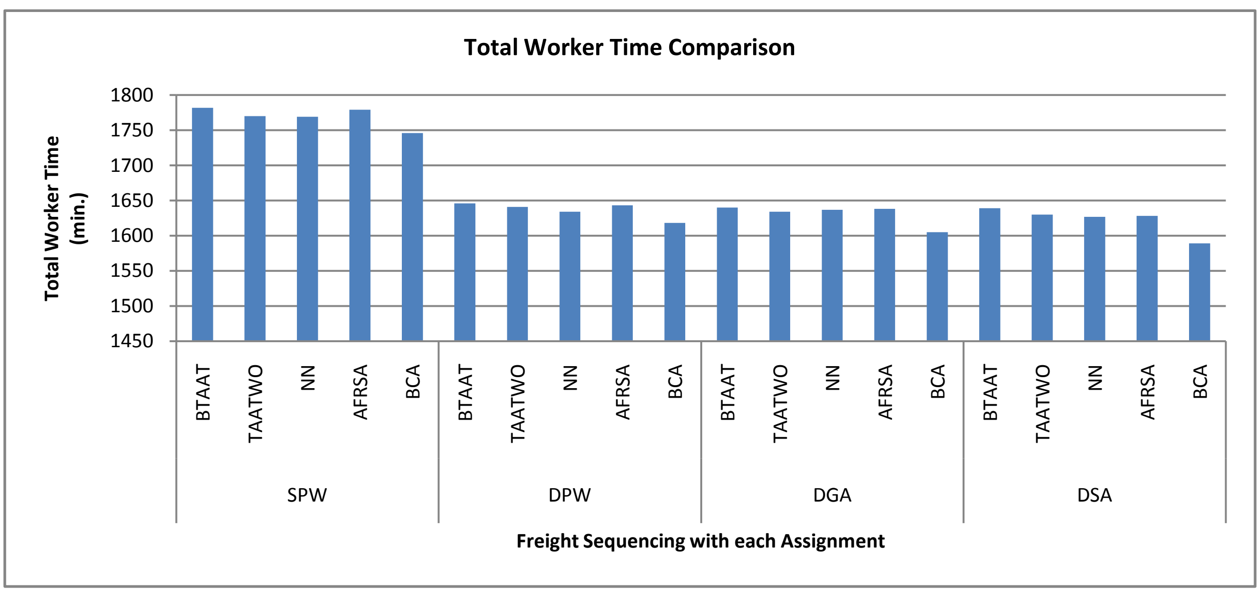


Figure 7.2 Total Worker Time Comparison from Data Set 1 for Simulation Experiment 2

Figure 7.3 illustrates the transfer time for freight sequencing with each door assignment from data set 1. As shown, changing the hub door assignment itself influences transfer time (make-span) but does not substantially reduce the transfer time (make-span) for the hub operations. The hub door assignment has less effect on changing the bottleneck worker time which determines the transfer time for the hub operations.

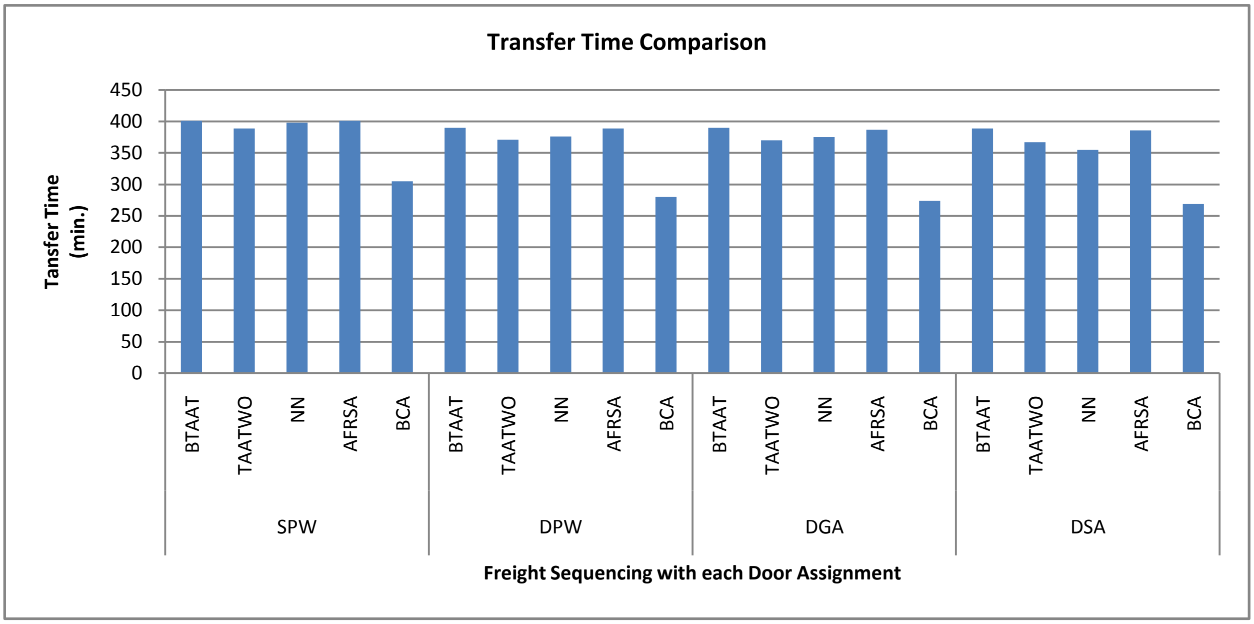


Figure 7.3 Total Transfer Time Comparison from Data Set 1 for Simulation Experiment 2

Figure 7.4 compares the average workload ratio for freight sequencing with each door assignment from data set 1. As shown, the BCA has the highest average workload ratio (more than 90%). The BCA reduces the workload of bottleneck worker and creates a workload more evenly distributed across all workers. The NN and TAATWO algorithm also reduce the workload of bottleneck worker and increase the average workload ratio since they also allow workers to share origin trailers.

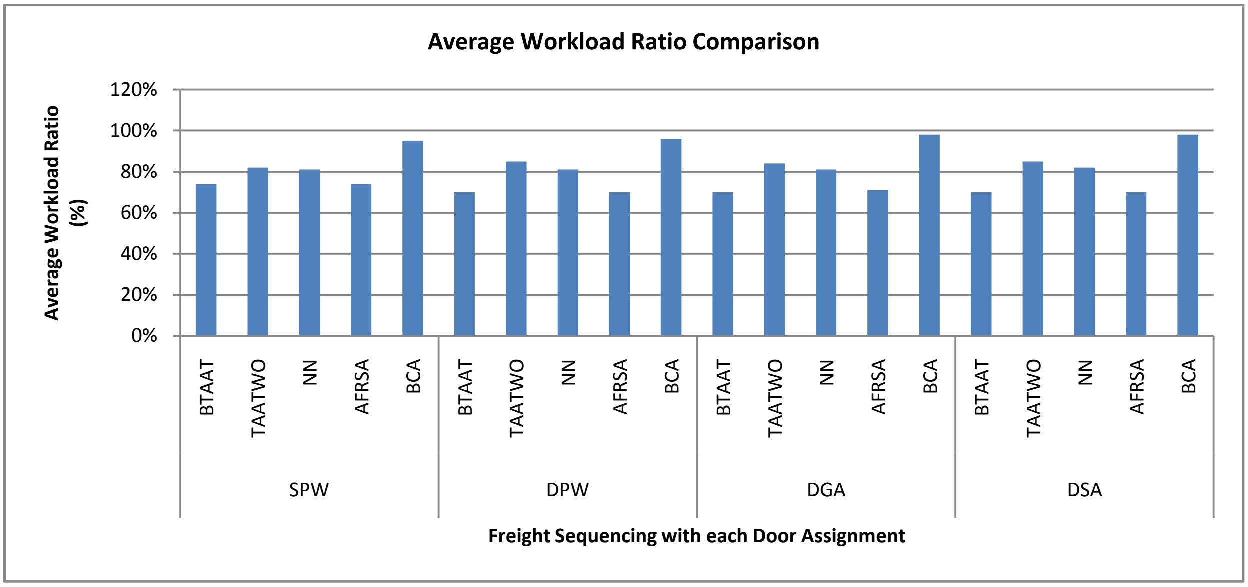


Figure 7.4 Average Workload Ratio from Data Set 1 for Simulation Experiment 2

The analysis of the door assignment and freight sequencing approaches for data set 2 and 3 are included in Appendix E. These comparisons also illustrate the same substantial improvements for dynamic door assignments over semi-permanent assignments.

The simulation results support the conclusions from Chapter 5. Adopting more efficient door assignment (such as dynamic door assignment with the Simulated Annealing methods) reduces the travel distance to transfer all shipments and reduces the labor time spent in the hub operations. The substantial reduction in the travel distance with the dynamic approach provides LTL carriers opportunities to reduce the time to transfer all shipments and also reduce the labor costs for the hub operations.

The impact of freight sequencing on the performance measures of hub operations is observed by comparing the simulation runs with the same door assignment. In Figure 7.3, the BCA freight sequencing substantially reduces on transfer time, compared to other freight sequencing approaches. In Figure 7.4, it is also shown that BCA has the highest average workload ratio (over 95%). This result shows that with the stochastic settings and congestion consideration in the simulation model, the Balance-and-Connect Algorithm reduces the make-span for hub operations, generates relatively balanced workloads, and offers opportunities to increase the customer service level for hub operations.

From comparing transfer time and average workload ratio for data set 2 and 3 (included in Appendix E), the NN and TAATWO freight sequencing approaches also reduce transfer time and increase average workload ratio, compared to BAAT and AFRSA. The NN and TAATWO approaches as with the BCA, allow sharing of origin trailers among workers, thus reducing the work time of bottleneck worker. The sharing effect is more obvious when there is a trailer that dominates the unloading time as in data sets 2 and 3. These trailers create a significant bottleneck worker with BTAAT and AFRSA, thus generating long transfer times for hub operations. The freight sequencing approaches which allow sharing of origin trailers among workers create more even workload among workers, reduce the bottleneck time, and thus reduce transfer time for hub operations.

7.5 Summary of Simulation Results

In this chapter, hub operations are modeled and analyzed using discrete event simulation, and a detailed simulation study is conducted for the hub operations. The solutions from the door assignment and freight sequencing approaches in previous chapters are evaluated using the simulation model with dynamic and stochastic features. By conducting simulation experiments for single worker and multiple workers, the effects of the door assignment and freight sequencing approaches are analyzed. The simulation results suggest that adopting dynamic door assignments (i.e., DPW, DGA and DSA) substantially reduce the total distance traveled by hub workers and the total worker time in the hub operations. Efficient freight sequencing approaches (i.e., BCA, NN and TAATWO) reduce the total distance, total worker time, and the transfer time (make-span) of the hub operations. The simulation results demonstrate that the performance measures of hub operations are improved by best combination of door assignment and freight sequencing approaches in a dynamic and stochastic environment.

Chapter VIII

Conclusions and Future Work

The objective of this research is to improve the efficiency of cross-dock operations in less-than-truckload hubs. The trailer-to-door assignment problem and the freight sequencing problem are addressed for the LTL hub operations. The following sections summarize the results of this research, describe the implementation issues, and provide recommendations for future work.

8.1 Summary and Conclusions

The trailer-to-door assignment problem is modeled as a Quadratic Assignment Problem (QAP). Two trailer-to-door assignment approaches are presented and compared: the *semi-permanent layout* and the *dynamic layout*. With a *semi-permanent layout,* the assignment of destination trailers to dock doors is developed on a periodic basis and origin trailers are assigned to dock doors on a nightly basis. With a *dynamic layout,* all trailers are assigned to dock doors on a nightly basis taking into account actual shipment information. Since the underling QAP structure of the trailer-to-door assignment problem is NP hard, heuristics methods are explored to solve the trailer-to-door assignment problem. The pair-wise exchange method is used for both the *semi-permanent layout* and *dynamic layout*. In an addition, a simulated annealing algorithm and genetic algorithm are explored for the *dynamic layout*. The simulated annealing algorithm performed well, and the preferred parameters are determined through experiment for the simulated annealing. In a case study, several data sets from LTL carriers are used to explore the solution approaches. From the numeric results, the *dynamic layout* out performs the *semi-permanent layout* for the total distance in all cases. Among the three heuristics for the *dynamic layout*, simulated annealing and genetic algorithm out perform the pair-wise exchange method. For the smaller sized problems, both the simulated annealing algorithm and genetic algorithm found the optimal solution. As the problem size increases, simulated annealing out performs genetic algorithm, but with longer computational time. The results of the case study demonstrate the substantial improvements possible from using a *dynamic layout* for the trailer-to-door assignment problem in the LTL hub operations. The *dynamic layout* provides the opportunity to substantially reduce the worker travel distance and labor time during the hub operations, thus improving the efficiency and reducing the labor cost of hub operations.

The freight sequencing problem (FSP) is modeled as a Rural Postman Problem (RPP). Three algorithms are presented for the FSP for a single worker: Trailer-at-a-Time, Nearest Neighbor, and Balance-and-Connect Algorithm for a single worker. A case study shows the Balance-and-Connect Algorithm out-performing TAAT and NN for the total travel distance, total time, bottleneck distance, and transfer time. For the FSP for multiple workers, the Balance-and-Connect Algorithm for multiple workers and Assign First Route Second Algorithm are investigated. Those algorithms compared to the Balanced Trailer-at-a-Time and other algorithms in the literature. The case study results show the Balance-and-Connect Algorithm for multiple workers out performs the other algorithms to reduce the total travel distance, transfer time, total labor time and bottleneck distance. The results illustrate the substantial opportunity to improve the efficiency and reduce labor cost in hub operations.

A simulation study is conducted in this research for the hub operations. The solutions approaches from the trailer-to-door assignment problem and freight sequencing problem with static and deterministic assumptions are evaluated in the simulation model with stochastic and dynamic features. By conducting simulation experiments, the impacts of different door assignment and freight sequencing rules are analyzed. The improvements of performance measures of hub operations from adopting simulated annealing door assignment and the Balance-and-Connect freight sequencing are significant in the simulation model. Adopting simulated annealing door assignment and the Balance-and-Connect freight sequencing can substantially reduce the bottleneck time (make-span), total worker time and total distance for the hub operations, thus improving the efficiency of hub operations.

This research makes the following main contributions:

- Extensive comparison between *semi-permanent layout* and *dynamic layout* for the trailer-to-door assignment problem;
- Simulated annealing and genetic algorithm for the *dynamic layout* for the trailer-to-door assignment problem;

- Model of the freight sequencing problem for a single worker as a Rural Postman Problem (RPP) and the freight sequencing problem for multiple workers as *k* Rural Postman Problem (*k* RPP);
- Balance-and-Connect Algorithm for the freight sequencing problem for a single worker and for multiple workers;
- Assign First Route Second Algorithm for the freight sequencing problem for multiple workers; and
- Discrete event simulation model to evaluate solution approaches for the trailer-at-a-door assignment problem and the freight sequencing problem in stochastic and dynamic environments.

The primary contribution of this research is the integer programming models and two heuristics for the freight sequencing problem (based on RPP) for a single worker and for multiple workers. This is first known the application of RPP for the FSP. The research also develops efficient heuristics based on the RPP formulations to solve the freight sequencing problem. The new modeling and solution approaches for hub operations raise implementation issues and the directions for future research.

8.2 Implementation Issues

The *dynamic layout* for the trailer-to-door assignment problem takes advantage of recent advances in information technology to assign all trailers to dock doors on a nightly basis. The dynamic approach reflects the changes in daily shipment characteristics in the hub operations. This approach relies on information of shipments from all customer service centers to obtain the assignments for trailers to dock doors before the hub operations. Most LTL carriers use a warehouse management system (WMS) and transportation management system (TMS) for their network operations. The recent development of Radio Frequency Identification (RFID) technology also makes it possible to capture and transmit shipments information along the LTL network. The improved visibility of shipment flow through WMS and TMS platform and RFID technology provides an opportunity to implement the dynamic layout to improve the hub operations.

The costs of implementing *dynamic layout* for the trailer-to-door assignment problem in the hub operations, however, must be justified carefully. The costs of implementing dynamic layout includes the cost of adopting information technology for the hub operations, the cost of managing new dock door assignment nightly, and the cost of potential misdirected freight as the daily changed dock door assignment (Brown, 2003). In this research the dynamic layout with pair-wise exchange, simulated annealing and genetic algorithm provide the trailer-to-door assignment solutions with reasonable computational time (within an hour) thus providing sufficient time for hub managers to finalize layouts before the hub operations. Adapting scanner technology when dock workers transfer shipments also reduce the probability of potential misdirected freight due to the dynamic changes of dock doors. If the time and labor savings due to the improvements from the dynamic layout are significant enough to overcome the costs of implementing the dynamic layout, the dynamic layout provides an attractive approach for the trailer-to-door assignment problem in the hub operations.

The Balance-and-Connect Algorithm for the freight sequencing problem (FSP) provides another opportunity for the LTL carriers to improve the efficiency and reduce labor costs in the hub operations. The solution from the Balance-and-Connect Algorithm for the FSP for multiple workers, however, raises other implementation issues. The Balance-and-Connect Algorithm for multiple workers reduces the transfer time for all shipments since it allows sharing the origin trailers among hub workers. While sharing of origin trailer offer the opportunity for balanced workload across each worker and the shortest make-span for hub operations, the solution might result in substantial congestions when multiple workers try to unload the same origin trailer at the same time. If the number of workers assigned to an origin trailer is restricted to reduce the potential congestion at the origin trailer, an additional constraint should be added to the FSP model and the Balance-and-Connect Algorithm for multiple workers should be amended accordingly. The Amended Balance-and-Connect Algorithm for multiple workers is discussed in Appendix C.

The Balance-and-Connect Algorithm requires that all trailers are available at the beginning of hub operations. Since the sequences of shipments transferred by hub workers are determined before the start of hub operations, any significant delay of trailer arrivals is difficult for the BCA to address. On the other hand, the Nearest Neighbor (NN) algorithm provides flexibility ith delays in trailer arrivals during at hub operations. The NN can be implemented when the arriving trailers are delayed and the pre-determined sequences of shipments from the BCA do not work.

Another implementation issue for the Balance-and-Connect for multiple workers is that the unloading process is constrained by precedence due to the order that the shipments are physically placed in the trailer. For example, if shipment 2 is physically placed in the trailer behind shipment 1, then in the shipment sequence, shipment 1 should be transferred before shipment 2 is transferred since it is not practical for the worker to take shipment 2 out of trailer before transferring shipment 1. For the solution approaches that do not allowing sharing of origin trailers, such as Balanced Trailer-at-a-time and Assign First and Route Second Algorithm, the precedence constraints are easily ensured. With Balance-and-Connect Algorithm for multiple workers, however, the precedence constraints may not be ensured without adjustment.

8.3 Future Work

This research addresses two operational decision problems in the LTL hub operations: The trailer-to-door assignment problem and freight sequencing problem. The trailer-to-door assignment problem and freight sequencing problem are interrelated problems in the sense that the impact of the trailer-to-door assignment is dependent on the effectiveness of the freight sequencing approach and the efficiency of the freight sequencing approach is dependent on the trailer-to-door assignment. While this research addresses both trailer-to-door assignment problem and the freight sequencing problem, the solutions are developed separately for each problem and two problems are solved sequentially. Ideally, the problems could be addressed simultaneously in the future work. This requires an integrated model to be developed for two problems and the solution approaches to be developed to solve two problems simultaneously.

In this research, the number of dock doors was assumed to be greater than the number of total trailers. While this assumption simplifies the modeling and solution approaches, it is not realistic when in some cases the number of total trailers is greater than the number of doors available. An area for future work on the trailer-to-door assignment problem is to develop an assignment of trailers to doors as they arrive at the hub. For the dynamic layout, other improvement techniques (such as tabu search and ant colony optimization) could be also used to solve the trailer-to-door assignment problem. Additional heuristics from the RPP and TSP literature may be explored . Also, the Balance-and-Connect Algorithm could be further investigated to address the potential congestions of sharing origin trailers and the precedence constraint issue.

In this research, reduction in total time (travel time plus the unloading and loading time) for the hub operations is not as significant as reductions in total travel distances, travel time and bottleneck time through the *dynamic layout* for the trailer-to-door assignment problem and through the Balance-and-Connect for the freight sequencing problem. The solution approaches in this research are focusing on reducing the travel distances and travel times for both the bottleneck worker and other workers. The unloading and loading time, however, are considered as fixed costs thus the improvement for the total time is limited by these fixed costs. In the future, research could also focus on improving the efficiency of the loading and unloading processes in hub operations.

REFERENCES

[1] Aarts, E., Korst, J., *Simulated Annealing and Boltzmann Machines*, Wiley, 1989.

[2] Adams, W.P., Guignard M., Hahn P.M., and Hightower W.L., " A Level-2 Reformulation–Linearization Technique Bound for the Quadratic Assignment Problem," *European Journal of Operational Research*, 180 (3), pp. 983-996, 2007.

[3] Adams, W.P., and Sherali, H.D., "Linearizarion Strategies for a Class of Zero-One Mixed Integer Programming Problems," *Operations Research*, 38(2), pp. 217-226, 1990.

[4] Ahuja, R.K., Orlin, J.B., and Tivari, A., "A Greedy Genetic Algorithm for the Quadratic Assignment Problem," Working Paper, MIT, 1995.

[5] Ball, M. O., and Magazine, M.J., "Sequencing of Insertions in Printed Circuit Board Assembly," *Operations Research*, 36(2), pp.192-201, 1988.

[6] Bartholdi, J. J., and Gue, K.R., "Reducing Labor Costs in an LTL Crossdocking Terminal," *Operations Research*, 48(6), pp. 823-832, 2000.

[7] Bazaraa, M.S., and Sherali, H.D., "Benders' Partitioning Scheme Applied to a New Formulation of the Quadratic Assignment Problem," *Naval Research Logistics Quarterly*, 27, pp. 29-41, 1980.

[8] Bebavent, E., Campos, V., Corberan, A., and Mota, E., "The Capacitated Arc Routing Problem: A Heuristic Algorithm," *Questiio*, 14(1-3), pp.107-122, 1990.

[9] Bennet, B., and Gazis, D., "School Bus Routing by Computer," *Transportations Research*, 6, pp. 317, 1972.

[10] Bermudez, R., and Cole, M.H., “A Genetic Approach to Door Assignment in Breakbulk Terminals,” Working Paper, University of Arkansas, Fayetteville, Department of Industrial Engineering, 2002.

[11] Bihr, R. A., “A Conceptual Solution to the Aircraft Gate Assignment Problem Using 0,1 Linear Programming,” *Computers and Industrial Engineering*, 19(1-4), pp. 280-284, 1990.

[12] Bodin, L.D., and Kursh, S.J., “A Computer – Assisted System for the Routing and Scheduling of Street Sweepers,” *Operations Research*, 26, pp. 525-537, 1978.

[13] Bozer, Y.A., and Carlo, H.J., “Optimizing Inbound and Outbound Door Assignments in Less-than-Truckload Crossdocks,” *IIE Transactions*, 40, pp. 1007-1018, 2008.

[14] Brown, A.M., “Improving the Efficiency of Hub Operations in a Less-than-Truckload Distribution Network,” MS thesis, Virginia Tech, 2003.

[15] Burkard, R.E., “Quadratic Assignment Problem,” *European Journal of Operation Research*, 15, pp. 283-289, 1984.

[16] Burkard, R.E., and Rendl, F., “A Thermodynamically motivated simulation Procedure for Combinatorial Optimization Problems,” Report 82-3, MIT, 1983.

[17] Chardaire, P., and Lutton. J.L. “Using Simulated Annealing to Solve Concentrator Location Problems in Telecommunication Networks,” in Rene V.V. Vidal (Ed.) *Applied Simulated Annealing*, Springer-Verlag, 1993.

[18] Connolly, David T., “An Improved Annealing Scheme for the QAP,” *European Journal of Operation Research*, 46, pp. 93-100, 1990.

[19] Christofides, N., “The Optimal Traversal of a Graph”, Omega 1(6), pp.719-732, 1973.

[20] Christofides, N., “Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem”, Report N388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, 1976.

[21] Christofides, N., Campos, V., Corberan, AQ., and Mota, E., “An Algorithm for the Rural Postman Problem on a Directed Graph”, *Mathematical Programming Study*, 26, pp. 155-166, 1986.

[22] Daganzo, Carlos F., *Logistics Systems Analysis*, Fourth Edition, Springer, 2005.

[23] Dickey, J.W., and J.W Hopkins “Campus Building Arrangement Using TOPAZ,” *Transportation Research*, 6, pp. 59-68, 1972.

[24] Duman, Ekrem., “The Quadratic Assignment Problem in the Context of the Printed Circuit Board Assembly Process,” *Computers and Operations Research*, 34, pp.163-179, 2007.

[25] Edmonds, J., and Johnson, E.L. , “Matching, Euler Tours and the Chinese Postman,” *Mathematical Programming*, 5, pp. 88-124, 1973.

[26] Eglese, R.W. “Routing Winter Gritting Vehicles”, *Discrete Applied Mathematics*, 48(3), pp. 231-244, 1994.

[27] Feitler, J. N., Corsi, T. M., and Curtis, M. G., “Strategic and Performance Changes Among LTL Motor Carriers: 1976-1993” *Transportation Journal*, 37(4), pp. 5-12, 1998.

[28] Frederickson, G.N., Hecht, M.S., and Kim, C.E. , “Approximation Algorithms for some Routing Problems,” *SIAM J. Computing*, 7 (2), pp.178-193, 1978.

[29] Gidas, B., "Non Stationary Markov Chains and Convergence of the Annealing Algorithm," *Journal of Statistical Physics*, 39(1-2), 1985.

[30] Gilmore, P.C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment problem," *SIAM Journal of Applied Mathematics*, 10, pp. 210-232, 1962.

[31] Goldberg, D., Genetic Algorithms in Search, *Optimization and Machine Learning*, Kluwer Academic Publishers,1989.

[32] Guan, M., "Graphic Programming Using Odd and Even Points," *Chinese Mathematics*, 1, pp. 273-277, 1962.

[33] Gue, Kevin R., "The Effects of Trailer Scheduling on the Layout of Freight Terminals," *Transportation Science*, 33(4), pp. 419-428, 1999.

[34] Gue, Kevin R., "Freight Terminal Layout and Operations," Ph.D. thesis, Georgia Institute of Technology, 1995.

[35] Haslam, E., and Wright, J.R., "Application of Routing Technologies to Rural Snow and Ice Control," *Transportation Research Record*, 1304, pp. 202-211, 1991.

[36] Heider, C.H., "A Computationally Simplified Pair Exchange Algorithm for the Quadratic Assignment Problem," Paper No. 101, Center for Naval Analyses, Arlington, VA, 1972.

[37] Hertz, A., Laporte, G., and Nanchen, P., "Improvement Procedures for the Undirected Rural Postman Problem", *INFORMS Journal on Computing*, 11, pp. 53-62, 1999.

[38] Hertz, A., Laporte, G., and Mittaz, M., "A Tabu Search Heuristic for the Capacitated Arc Routing Problem," *Operations Research*, 48(A), 2000.

[39] Holland, John H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[40] Kelton, W.D., Sadowski, R.P., and Sturrock, D.T., *Simulation with Arena*, 3rd edition, McGraw Hill, 2007.

[41] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.M. , "Optimization by Simulated Annealing," *Science*, 220, pp. 671-680, 1983.

[42] Kochhar, J., Foster, B., and Heragu, S., "HOPE: A Genetic Algorithm for Bin Packing," *Computer Math*, 63 (September), pp. 203-214, 1998.

[43] Koopmans, T.C., and Beckmann, M.J. "Assignment Problems and the Location of Economic Activities," *Econometrica*, 25, pp. 53-76, 1957.

[44] Lenstra, J.K., and Kan, R., "On General Routing Problems," *Networks*, 6, pp 273-280, 1976.

[45] Lin, S., and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research*, 21, pp. 498-516, 1973.

[46] Meller, R.D., and Bozer, Y.A., "A New Simulated Annealing Algorithm for the Facility Layout Problem," *International Journal of Production Research*, 34(6), pp.1675- 1692, 1996.

[47] Mitchell, M., An Introduction to Genetic Algorithms, Cambridge, MIT Press, 1997.

[48] Mangoubi, R.S., and Mathaisel, F.X., "Optimizing Gate Assignment at Airport Terminals," *Transportation Science*, 19(2), pp. 173-188, 1995.

[49] Napolitano, M., “Warehousing and Distribution Center Management: The Cross-dock revolution – are you in or out?” *Logistics Management*, April 1, 2008.

[50] Nissen, V., “Solving the Quadratic Assignment Problem with Clues from Nature,” *IEEE Transactions on Neural Networks*, 5(1), pp.66-72, 1994.

[51] Nissen, V., and Paul, H., “A Modification of Threshold Accepting and its Application to the Quadratic Assignment Problem,” *OR Spektrum*, 17, pp. 205-210, 1995.

[52] Peck, Ken E., “Operational Analysis of Freight Terminal Handling Less Than Container Load Shipments,” Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1983.

[53] Pearn, W.L., “Augment-Insert Algorithms for the Capacitated Arc Routing Problem,” *Computer Operations Research*, 18(2), 189-198, 1991.

[54] Rasmussen, R., “QAP – Not So Hard In Spreadsheets,” *Omega*, 35, pp. 541-552, 2007.

[55] Reeves, C.R., Modern Heuristic Techniques For Combinatorial Problems, John Wiley & Sons, Inc. New York, 1993.

[56] Rosen, K. H. (Editor), and Michaels, J.G. (Editor)., *Handbook of Discrete and Combinatorial Mathematics*, Boca Raton, Florida, CRC Press, 2000.

[57] Sahni, S., Gonzalez, T., “P-complete approximation problems,” *Journal of the Association for Computing Machinery*, 23, pp. 555-565, 1976.

[58] Striker, R., “Public Sector Vehicle Routing: The Chinese Postman Problem,” M.Sc. Dissertation, Department of Electrical Engineering, MIT, Cambridge, 1970.

[59] Taillard, E.D., "Robust Taboo Search for the Quadratic Assignment problem," *Parallel Computing*, 17, pp. 443-455, 1991.

[60] Tang, A.P., and Ma, Y., "The Intrastate Deregulation and the Operating Performance of Trucking Firms," *American Business Review*, 20(1), pp. 33-42, 2002.

[61] Tavakkoli-Moghaddain, R., and Shayan, E., "Facilities Layout Design by Genetic Algorithms," *Computers & Operations Research*, 22(1), pp. 73-83, 1994.

[62] Thonemann, U.W., and Bolte, A., "An Improved Simulated Annealing Algorithm for the Quadratic Assignment Problem," Technical Report, Department of Production and Operations Research, University of Paderborn, Allemagne, Germany, 1994.

[63] Tsui, L.Y., and Chang, C., "A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards," *Computer and Industrial Engineering*, 19(1-4), pp. 309-312, 1990.

[64] Tsui, L. Y., and Chang, C., " An Optimal Solution to a Dock Door Assignment Problem," *Computer and Industrial Engineering*, 23(1-4), pp. 283-286, 1992.

[65] Ulusoy, G., "The Fleet Size and Mix Problem for Capacitated Arc Routing", *European Journal Operations Research*, 22(3), pp. 329-337, 1985.

[66] Vidal, Rene V.V.(Editor) Applied Simulated Annealing, Springer-Verlag, 1993.

[67] Wilson, R. A., "Economic Impact of Logistics," in G. Don Taylor (Ed.) *Logistics Engineering Handbook*, CRC Press, 2008.

Appendix

Appendix A

Pair-wise Exchange and Simulated Annealing C++ Code

```
float Pairwise()
{
	int j;

	float CurrentDistance=0;
	for( j=0;j<NumHubDoors;j++)
	{
		Door[j]=OriginalDoor[j];
	}
	for(j=0;j<=TotalDoors;j++)
	{
		Stack[j]=0;
	}
	for(j=0;j<NumHubDoors;j++)
		for(int k=0;k<NumHubDoors;k++)
		{
			CurrentDistance=CurrentDistance+2*TotalHU[j][k]*Dist[Door[k]][Door[j]];
		}

	int TempDoor;
	float NewDist;


Next:
	for(j=0;j<NumHubDoors;j++)
		for(int k=j+1;k<NumHubDoors;k++)
		{
			TempDoor=Door[j];
			Door[j]=Door[k];
			Door[k]=TempDoor;
			NewDist=0;
			for(int p=0;p<NumHubDoors;p++)
				for(int l=0;l<NumHubDoors;l++)
				{
					NewDist=NewDist+2*TotalHU[p][l]*Dist[Door[l]][Door[p]];
				}
			if(NewDist<CurrentDistance)
			{
				CurrentDistance=NewDist;
				v2.push_back(CurrentDistance);
				goto Next;
			}
			else
			{
				Door[k]=Door[j];
				Door[j]=TempDoor;
			}
		}
```

```
	for(int k=0;k<TotalTrailers+1;k++)
	{
		if(Type[k]==2||Type[k]==3)
		{
			Stack[Door[k]]=1;
		}
	}

ofstream outfile("DoorAssignmentOutput.in",ios::app);
				outfile<< "Pairwise Door Assignments  "<<endl;
				for(j=0;j<NumHubDoors;j++)
					{
					if(Type[j]==1)
					{
						outfile<<"Origin Trailer "<<ID[j]<<" is assigned to door
"<<Door[j]<<endl;
					}
					if(Type[j]==2)
					{
						outfile<<"Destination "<<ID[j]<<" is assigned to door
"<<Door[j]<<endl;
					}
					if(Type[j]==3)
					{
						outfile<<"Miscellaneous Destination is assigned to door
"<<Door[DestDoors-1]<<endl;
					}
					}
				outfile<<"The Current Distance for Pairwise exchange is"<<"
"<<CurrentDistance<<endl;
				outfile.close();

ofstream ot("ObjectiveValue-PA.in",ios::app);

		list<int>::iterator i;

		 for ( i=v2.begin(); i!= v2.end(); i++)
			{
				ot<< (*i) <<endl;
			}
	return CurrentDistance;
}


float SimulatedAnnealing()
{
	// initial Door assignments
	int j;

	float CurrentDistance=0;
	float CurrentTime=0;
	for( j=0;j<NumHubDoors;j++)
	{
		Door[j]=OriginalDoor[j];
	}
	for(j=0;j<=TotalDoors;j++)
```

```
            {
                        Stack[j]=0;
            }
            for(j=0;j<NumHubDoors;j++)
                        for(int k=0;k<NumHubDoors;k++)
                        {
                                    CurrentDistance=CurrentDistance+2*TotalHU[j][k]*Dist[Door[k]][Door[j]];
                        }

            int TempDoor;
            float NewDist;
            float NewTime;

  //Pairwise();
  //CurrentDistance=Pairwise();

float T = CurrentDistance/40;
int AcceptCount = 0;
int AcceptCountIsZero = 0;
int AttemptCount = 0;
float Diff = 0;

double rnd;
int integerRnd;

// improve the Door assignments by Pair-wise Exchanged Simulated Annealing algorithm

while ( T>1)
{
            while (AcceptCount < 500 && AttemptCount < 500)
            {

            for(int j=0;j<NumHubDoors;j++)              // Pair-wise Exchange the door

                        for(int k=j+1;k<NumHubDoors;k++)
                        {

                                    TempDoor=Door[j];
                                    Door[j]=Door[k];
                                    Door[k]=TempDoor;
                                    NewDist=0;
                                    for(int p=0;p<NumHubDoors;p++)
                                                for(int l=0;l<NumHubDoors;l++)
                                                {
                                                            NewDist=NewDist+2*TotalHU[p][l]*Dist[Door[l]][Door[p]];

                                                }
        Diff = NewDist - CurrentDistance;
                        // Generate Random Number

                          // rand() returns a value from 0 to 32767
                                                rnd = 10000. / rand();
//                                              cout<<"first: "<<rnd<<endl;
                                                // integer part of quotient
                                                integerRnd = rnd;
                                                // every random number in the form of 0.xxx
```

```
                        rnd -= integerRnd;
//                      cout<<"get a new rnd: "<<rnd<<endl;



                float RandomNumber = rnd;
                float ExponentialNumber = exp(-Diff/T);

                if(Diff < 0)
                {
                        printf("\Diff is < 0 ! Accept new distance !\n");
                        cout << CurrentDistance<<endl;
                        v.push_back(CurrentDistance);

                        cout << NewDist<<endl;
                        //cout << RandomNumber<<endl;
                        //cout << "Aceept Probability is "<<ExponentialNumber<<endl;
                        cout << "Accept times is  "<<AcceptCount<<endl;
                        cout<< "Temperature is  " <<T<<endl;
                        CurrentDistance=NewDist;
                        AcceptCount++;

                }
                else if ( Diff >0 && RandomNumber < ExponentialNumber )
                {
                        printf("\Diff is > 0 ! But we still accept new distance !\n");
                        cout << CurrentDistance<<endl;
                        v.push_back(CurrentDistance);

                        cout << NewDist<<endl;
                        cout << RandomNumber<<endl;
                        cout << "Aceept Probability is "<<ExponentialNumber<<endl;
                        cout << "Accept times is  "<<AcceptCount<<endl;
                        cout<< "Temperature is  " <<T<<endl;
                        CurrentDistance=NewDist;
                        AcceptCount++;

                        //goto Next;

                }
                else
                {
                        printf("\Diff is > 0 and we don't accept new distance! \n");
                        cout << CurrentDistance<<endl;
                        cout << NewDist<<endl;
                        cout << RandomNumber<<endl;
                        cout << "Aceept Probability is "<<ExponentialNumber<<endl;
                        Door[k]=Door[j];
                        Door[j]=TempDoor;
                        AttemptCount++;
                        cout<<"Attempt times is  "<<AttemptCount<<endl;
                        cout<< "Temperature is  " <<T<<endl;
                }

        }
```

```
            for(int k=0;k<TotalTrailers+1;k++)
            {
                        if(Type[k]==2||Type[k]==3)
                        {
                                    Stack[Door[k]]=1;
                        }
            }

            if (AcceptCount=0)
            {
                        AcceptCountIsZero = AcceptCountIsZero + 1;
            }
  }

T = 0.9*T;
printf("\Temperature reduced! \n");
cout<< "Temperature is  " <<T<<endl;
AcceptCount = 0;
AttemptCount = 0;
}
ofstream outfile("DoorAssignmentOutput.in",ios::app);
                                        outfile<< "SA Door Assignments  "<<endl;
                                        for(int j=0;j<NumHubDoors;j++)
                                                    {
                                                    if(Type[j]==1)
                                                    {
                                                                outfile<<"Origin Trailer "<<ID[j]<<" is assigned to door
"<<Door[j]<<endl;
                                                    }
                                                    if(Type[j]==2)
                                                    {
                                                                outfile<<"Destination "<<ID[j]<<" is assigned to door
"<<Door[j]<<endl;
                                                    }
                                                    if(Type[j]==3)
                                                    {
                                                                outfile<<"Miscellaneous Destination is assigned to door
"<<Door[DestDoors-1]<<endl;
                                                    }

                                                                outfile<<"The Current Distance for Simulated Annealing
is"<<" "<<CurrentDistance<<endl;
                                        outfile.close();

       ofstream ou("ObjectiveValue.in",ios::app);

                     list<int>::iterator i;

                      for ( i=v.begin(); i != v.end(); i++)
                              {
                                        ou << (*i) <<endl;
                              }
                                        ou.close();

return CurrentDistance;
}
```

Appendix B

AMPL model for the FSP

AMPL model for the FSP for a Single Worker

```
set DOOR;
set LINKS dimen 2;

#set Arcs within {DOOR, DOOR};

param dist{DOOR, DOOR} >=0; #door distances
param d{DOOR};
param f{LINKS};



var X{DOOR, DOOR} >=0; #origins assignments



minimize total_distance:
   #sum{i in DOOR, j in DOOR}dist[i,j]*X[i,j];

sum{ i in DOOR, j in DOOR}(dist[i,j]/232)*X[i,j]+sum{(i,j) in
LINKS}(1.46)*X[i,j];

subject to constr1{i in DOOR}: sum{k in DOOR}X[i,k] = d[i];

subject to constr2{i in DOOR}: sum{k in DOOR}X[k,i] = d[i];

subject to constr3{(i,j) in LINKS}: X[i,j]=f[i,j];
```

AMPL model for the FSP for *k* Workers

```
set K;
set DOOR;
set LINKS dimen 2;

#set Arcs within {DOOR, DOOR};

param dist{DOOR, DOOR} >=0; #door distances
param d{DOOR};
param f{LINKS};



var X{DOOR, DOOR, K} >=0 integer ; #origins assignments
var z >=0; #artificial variable
```

```
#var y{DOOR, K} >=0 binary;


minimize total_distance: z;

subject to constr1{k in K}: sum{ i in DOOR, j in
DOOR}(dist[i,j]/232)*X[i,j,k]+sum{(i,j) in LINKS}(1.46)*X[i,j,k]<=z;

subject to constr2{i in DOOR}: sum{k in K} sum{j in DOOR}X[i,j, k] = d[i];

subject to constr3{i in DOOR}: sum {k in K} sum{j in DOOR}X[j,i,k] = d[i];

subject to constr4{(i,j) in LINKS}: sum {k in K} X[i,j,k]=f[i,j];

#subject to constr6{i in DOOR, k in K, (i,j) in LINKS}:
X[i,j,k]=f[i,j]*y[i,k];

#subject to constr7{i in DOOR}: sum{k in K}y[i,k] <=2;

#subject to constr8{i in DOOR}: sum{k in K}y[i,k] >=1;

subject to constr5{k in K, i in DOOR}: sum{j in DOOR}X[i,j,k] - sum{j in
DOOR}X[j,i,k] = 0;
```

Appendix C

Illustrative Example for the Balance-and-Connect Algorithm

In this example, the algorithm is used to develop the freight unloading and loading sequence for using a small instance (2 origin trailers and 5 destination trailers) for a hub operation. Assume the optimal door assignment solution is determined using the approach from Chapter 4. The door assignment layout and required shipment flow is illustrated in Figure C.1. The Balance-and-Connect Algorithm is illustrated using the following steps:

Step 1: Setup the directed RPP network. Determine the degree for each node on the network;

First setup the directed RPP network. Figure A.C.2 represents graph $G_R = (N, R)$. R is the set of required arcs. The data of the number of required arcs is from the shipment flow data in the hub operations.

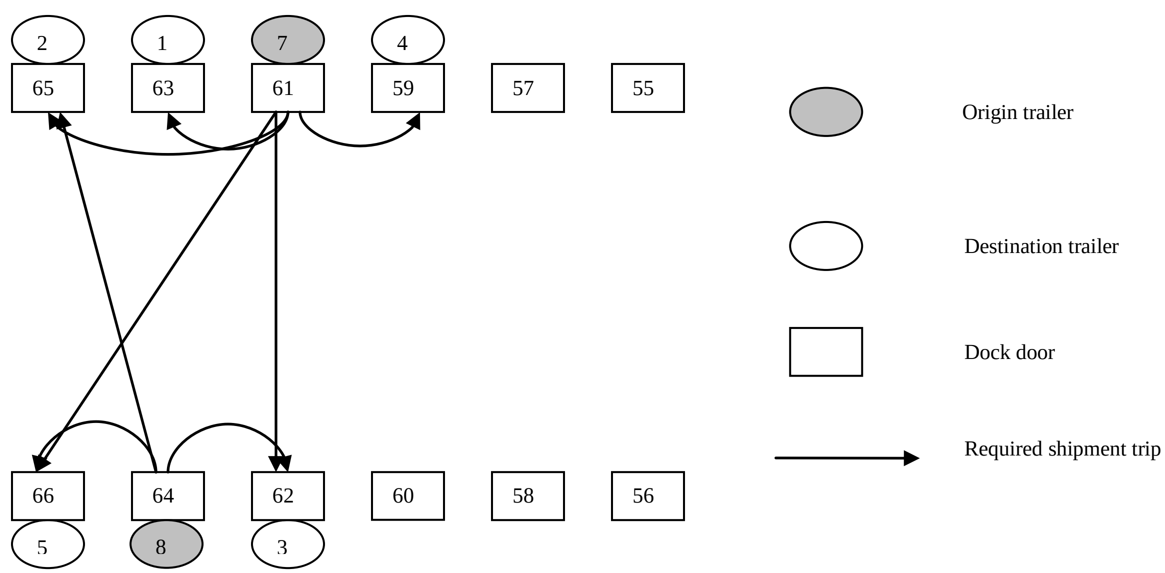


Figure C.1 Door Assignment Layout

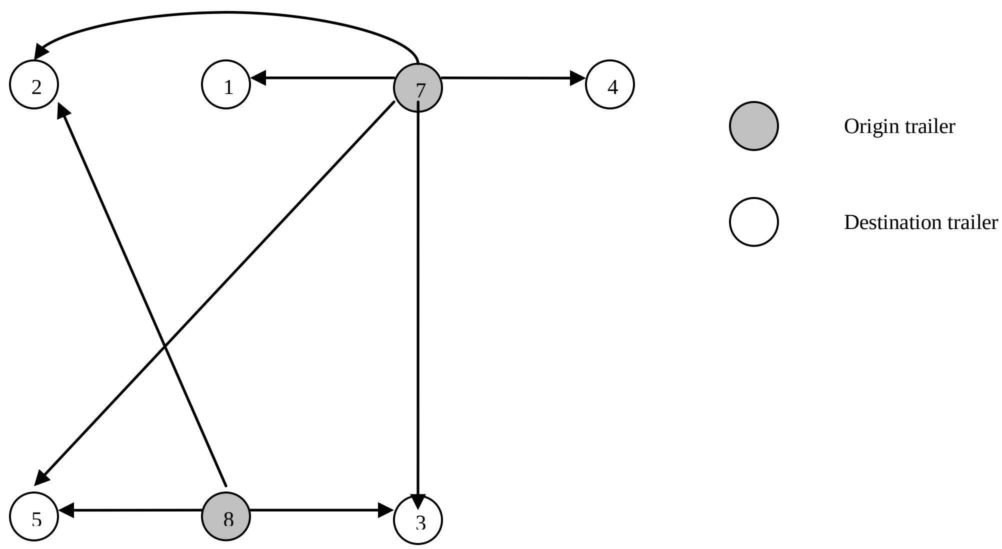


Figure C.2 Graph Representing of Required Shipment Movements

The d_i = degree (i) for each node $i \in N$ is computed and summarized in the Table C.1 :

Table C.1 Calculate Degree for Each Node in the Example

i	1	2	3	4	5	7	8
d_i	1	2	2	1	2	5	3

The d_i for node 3 (destination trailer door) is 2, meaning there have 2 required arcs flowing into node 3. The d_i for node 7 (origin trailer door) is 5, meaning there have 5 required arcs flowing out of node 7.

Step 2: Solve the relaxation.

Solve the minimum cost network flow problem:

Minimize $\sum_{(i,j)\in A} (c_{ij}/v)x_{ij} + \sum_{(i,j)\in R} (u_i + l_j)x_{ij}$

Subject to:

$$\sum_{i:(i,j)\in A} x_{ij} = d_i \quad \text{For all i = 1, 2,3,4,5, 7, 8}$$

$$\sum_{i:(j,i)\in A} x_{ji} = d_i \quad \text{For all i = 1, 2,3,4,5, 7, 8}$$

$$x_{ij} = f_{ij} \qquad \text{for all } (i,j) \in R$$

$$x_{ij} \geq 0, \quad \text{Integer}$$

The solutions (bold and dashed arcs) represent the required arcs and selected non-required arcs:

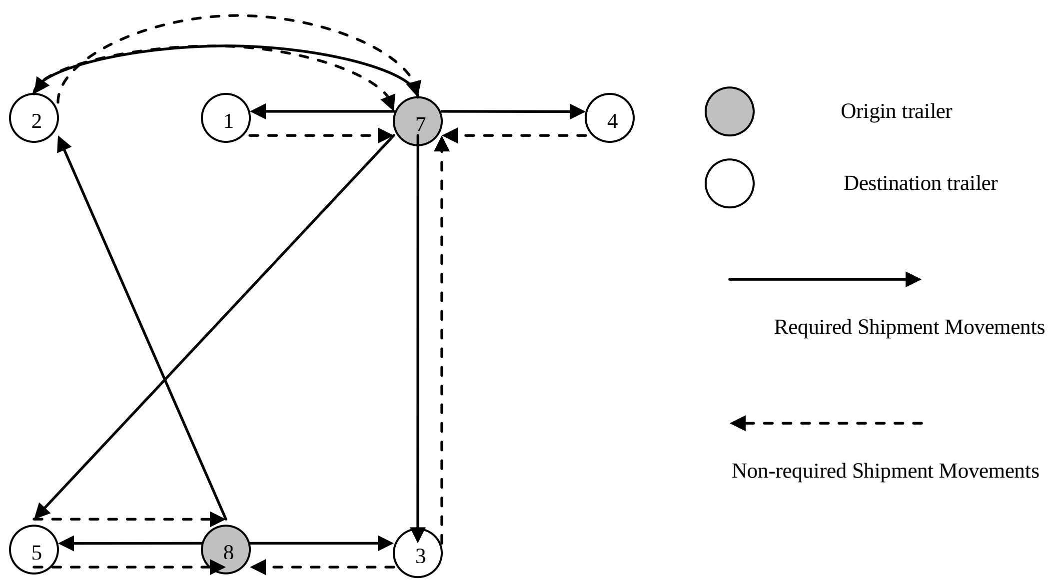


Figure C.3 Graph of Required Movements and Selected Non-Required Movements

Step 3: Check the connectivity of graph *G = (N, A)*.

The graph is checked for the connectivity using the connectivity test procedure.

a. Start from node 7:

L = {7}; K = {7}

b. Explore all reachable nodes from 7:

K = {7, 1, 2, 3, 4, 5}

c. Add node 1, 2, 3, 4, 5 to L;

L = {7, 1, 2, 3, 4, 5}

d. Delete node 7 from K and start to check the node 2:

K = {2, 3, 4, 5} (don't add node 7 to K since node 7 was already checked)

e. Repeat step d to check node 3, 4, 5:

K = {8}, L = {7, 1, 2, 3, 4, 5, 8}

f. Check node 8:

K = {}, L = {7, 1, 2, 3, 4, 5, 8}

g. Since K = {}, stop.

L covers all nodes so the graph is connected.

Step 4: Apply the Euler tour algorithm to find a sequence.

a. Find all cycles in the Graph:

The cycles $(7 \to 1 \to 7)$, $(7 \to 2 \to 7)$, $(7 \to 3 \to 7)$, $(7 \to 4 \to 7)$, $(7 \to 5 \to 8 \to 2 \to 7)$, $(8 \to 3 \to 8)$, $(8 \to 5 \to 8)$ are found in graph G.

b. Splice the cycles to form an Euler tour:

The above cycles are spliced to form an Euler tour:

$(7 \to 1 \to 7 \to 2 \to 7 \to 3 \to 7 \to 4 \to 7 \to 5 \to 8 \to 3 \to 8 \to 5 \to 8 \to 2 \to 7)$

Thus a Euler tour to transfer the shipments involves the following path for a worker: (7,1), (1,7), (7,2), (2,7), (7,3), (3,7), (7,4), (4,7), (7,5), (5,8), (8,3), (3,8), (8,5), (5,8), (8,2), (2,7).

Appendix D

Amended Balance-and-Connect Algorithm for FSP for *k* workers

The IP formulations for the FSP for k workers allow workers to take any arc (i, j). Required arcs from one node i can be assigned to different workers such that workers may share the same origin trailers. For example, for node 1 (origin trailer1) there are 2 shipments needed to be transfer to 2 destination trailers. To minimize the maximum completion time, the first shipment is assigned to worker 1 and the second one is assigned to worker 2. While sharing of origin trailer offer the opportunity for balanced workload across each worker and the shortest make-span for hub operations, the solution might result in substantial congestions when multiple workers try to unload the same origin trailer at the same time. It is not also desirable for the hub supervisor to assign too many workers to same origin trailer. If we restrict the number of workers that can be assigned to an origin trailer, an additional constraint should be added to the model and amend the Balance-and-Connect Algorithm. Solutions with workers sharing origin trailer nodes will be checked for acceptability (using pre-defined acceptable criterion, i.e., no more than 3 workers sharing one origin trailer). If the solution is not acceptable, new constraints are added (adjusting the solutions based on the sharing criterion) to the IP model and solve the model again. If the solution is acceptable, check the connectivity of each worker and connect the sub-tour if necessary. Finally find the sequence for each worker by applying Euler tour algorithm. The Amended BCA is summarized following steps and in Figure D.1.

Step 1: Solve the relaxation model (4.23) – (4.30) without constraint (4.29).

Step 2: Check the obtained solution with sharing acceptable criterion. If it is acceptable, go to Step 4; otherwise go to step 3;

Step 3: Add new constraint to the model based on adjusting sharing solution and go to Step 1;

Step 4: Check the connectivity. Perform minimum spanning tree algorithm to connect the sub-tour if necessary.

Step 5: Apply Euler tour algorithm to find the sequence for each worker.

To restrict the number of workers assigned to one trailer, the following constraints are considered to add to the model. Let define binary variable y_i^k where y_i^k = 1 if the k^{th} worker transfer shipment from node i, 0 otherwise. Let n be the number of workers which can share one origin trailer. The following constraints are added to the model:

$$x_{ij}^k \le f_{ij} y_i^k \quad \text{for all } i \in N\ k \in K \text{ and } (i,j) \in R \qquad \text{(D.1)}$$

$$\sum_k y_i^k = n \quad \text{for all } i \in N\ k \in K \qquad \text{(D.2)}$$

$$y_i^k \text{ Binary} \qquad \text{(D.3)}$$

The constraint (D.1) ensures that if y_i^k = 1 then $x_{ij}^k \le f_{ij}$ which means that if the k^{th} worker transfer shipment from node *i*, then the number of arcs that k^{th} worker taken from *i* to *j*, x_{ij}^k, is positive number and less than f_{ij}. If y_i^k = 0 then $x_{ij}^k = 0$. Constraint (D.2) ensures that only *n* workers can share one origin trailer. This *n* can be determined by the practices of hub operations. Notice that if *n* equals to the number of total workers in the hub operations, then this is exactly the BAC without any amendments. If *n* equals 1, with only one worker is assigned to one trailer, then this is similar to the AFRSA.

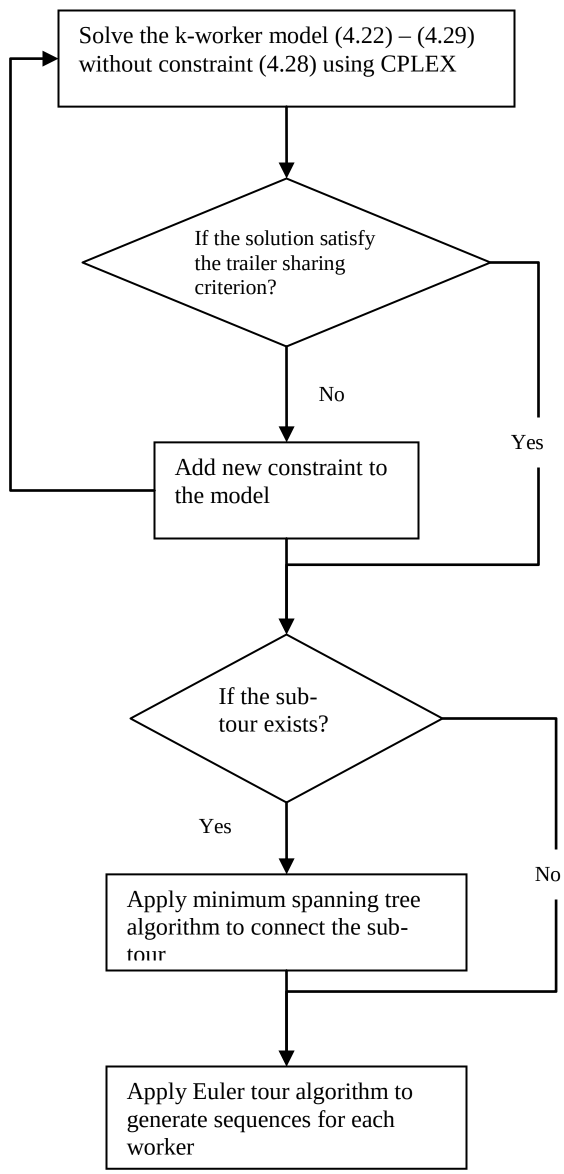


Figure D.1 Amended Balance-and-Connect Algorithm for *k* Workers Freight Sequencing

Appendix E

Simulation Model Details

The details of the simulation model are provided in this appendix. The data requirements, the simulation models, the ARENA model, the validation process and the simulation results are described.

E.1 Data Inputs of the Simulation Model

The primary inputs to the simulation model include the solutions from the models in previous chapters and the data for the hub layout and shipment flow. Details of data are described below.

1) Solutions from solution approaches for the trailer-to-door assignment and freight unloading and loading sequence problems, including:
    - The assignment of both origin and destination trailers to the hub doors.
    - The assignment of shipments to the hub worker and the sequence of unloading and loading for each shipment.

2) Data for the hub operations including:
    - The physical layout of hub. This physical layout information includes the shape of the hub, the physical position of hub doors, the total doors to be used to for hub operations, and the distance between the doors. In this simulation model, a rectangular hub with 33 hub doors located on both sides of the hub is used. The distance between hub doors is provided by a national LTL carrier.
    - Shipment data specifies the origin and destination of each shipment and the number of handling units. Trailer data specify the shipment details on each origin and destination trailer. These data are from the historical database from a national LTL carrier's hub operations. Table 7.1 provides a summary of shipment and trailer data.
    - Other data for the hub operations, such as worker's travel speed, unloading and loading time. The data are considered deterministic in the solution approaches for the trailer-to-door assignment and freight unloading and loading sequence problems. In

this simulation model the data are random variables that follow some specified distributions.

E.2 Simulation of Hub Operations

Detailed hub operations processes including trailer arrival and departure process, workers operations process, freight unloading and loading process are described in the following sections.

E.2.1 Trailer Arrival and Departure Process

1) Trailer Arrival

The simulation model simulates one night's activity at a LTL hub. Shipment transferring activities begin at 9:00 pm at the hub when the first origin trailer arrives the hub. Shipment transferring activities will end when all shipments are transferred to their destination trailers. In the simulation run for the dynamic arrivals of origin trailer, the actual arrival times of origin trailer are using random number within a time range.

If the number of hub doors is greater than the number of total trailers in the hub operations and all origin trailers arrive the hub before the transferring operations start, the simulation model will input the solutions from Chapter 5 to assign the doors to all origin and destination trailers. Since all trailers are available at the beginning of the hub operations, the simulation model will evaluate the solution from Chapter 6 as well as the solutions from trailer-at-a-time and nearest neighbor approaches for the freight unloading and loading sequence problem based on the stochastic settings for worker's unloading and loading time, and workers travel speed.

If some origin trailers arrive at the hub after the hub operations begin, then the late trailers will be assigned to the door according the solution from Chapter 5. Since late trailers are not available at the beginning of the hub operations, the solution from Chapter 6 is not applicable to assign the worker's freight unloading and loading sequence, trailer-at-a-time and/or nearest neighbor approaches are used for the freight unloading and loading sequence problem based on the stochastic settings for worker's unloading and loading time and workers travel speed.

2) Trailer Departure Process

Destination trailers are assigned to the hub doors by the door using the assignment solution from Chapter 5. A destination trailer departs the hub when the trailer is full. Destination trailers also departure the hub if no more shipment to that destination for the evening.

The trailer Arrival and Departure Process is summarized in the Figure E.1.

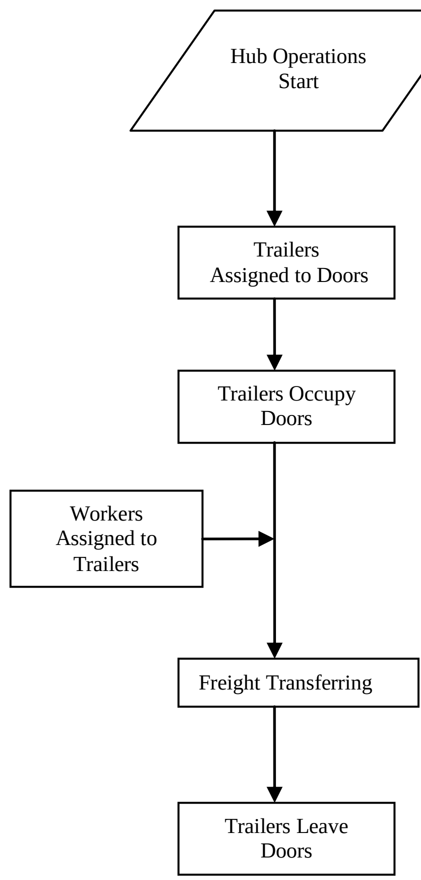


Figure E.1 Trailers Arrival and Departure Process

E.2.2 Hub Worker Operations

In Chapter 6, for a given number of hub workers, origin trailers are assigned to hub workers based on the balancing approach to have a quite similar workload for each worker. The more workers in the hub operations, the less the average worker time needed to complete all shipment transferring while the total worker's travel distance keeps no change. In the simulation model, the number of hub workers could be treated as a control parameter to evaluate the impact of different freight unloading and loading sequence approaches. The savings in terms of travel distance and time from efficient door assignment and freight sequencing approach can be translated in terms of labor savings.

1) The Arrival and Departure of Hub Workers

The number of hub workers present can vary over the night in the real hub operations. In this simulation model, we are assuming there are two hub workers at the beginning of the hub operation. The remaining hub workers will join the hub operations after some origin trailers arrive at the hub since some hub workers are actually the trailer drivers.

In this simulation model, the departure time for hub workers can vary also. In the simulation model, hub workers may depart before the end of hub operations. Hub workers leave after unloading and loading all of their assigned shipments.

In the Balance-and-Connect freight sequencing approach, multiple workers may be assigned to the same origin trailer and, congestion may occur at the origin trailer when multiple workers try to unload the same origin trailer. Congestion also happens at the destination trailers when multiple workers try to load to the same destination trailer.

2) Assign Hub Workers to Trailers

Hub workers are assigned trailers based on the solutions from previous chapters. If trailer-at-a-time and Nearest Neighbor freight sequencing approaches are used, the trailers assigned to the workers are from the "balanced" assignment solutions discussed in Chapter 6. Each trailer is associated with a single worker. No sharing of origin trailers exits. If Balance-and-Connect freight sequencing is used, each shipment is assigned to a worker based on the solutions from Balance-and-Connect freight sequencing. Each origin trailer is assigned with multiple workers.

The Hub workers Operations Process is summarized in Figure E.2.

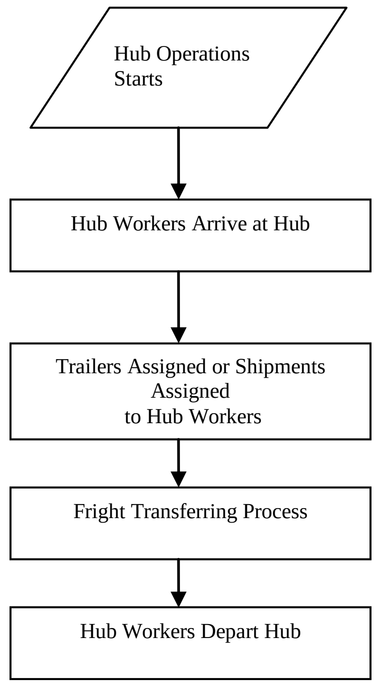


Figure E.2 Hub Workers Operations Process

E.2.3 Freight Transferring Process

In this research, a handling unit of a shipment is assumed to be a load that a hub worker can carry using a forklift. If a shipment contains multiple handling units, multiple trips are required for the hub worker to finish that shipment. In the simulation, when Balance-and-Connect freight sequencing is used, a shipment could be assigned to different workers.

1) Unloading Freight

The unloading activity starts when a hub worker driving a forklift enters an available origin trailer and ends when this hub worker leaves the origin trailer with one handling unit on the forklift. In the previous chapters, the unloading time is deterministic with the value of 0.74 minutes, which is an average number from the results of a time study in a hub. In this simulation model, however, the unloading time is estimated by uniform distribution UNIF(0.70 minutes, 0.78 minutes) with the mean of 0.74 minutes for the unloading time.

The simulation model assumes that for each worker, unloading time for transferring different handling units will follow the same distribution, regardless of freight characteristics, such as weight. When a hub worker is unloading an origin trailer, other workers wanting to unload the same origin trailer must wait in queue for the trailer to become available.

2) Loading Freight

The loading activity starts when a hub worker driving a forklift with one handling unit enters into an available destination trailer, ends when this hub worker leaves the destination trailer with the empty forklift. The loading activity in the simulation model is estimated by the uniform distribution UNIF(0.70 minutes, 0.78 minutes) with the mean of 0.74 minutes for the loading time.

As with the origin trailers, a hub worker can not load freight onto a destination trailer if another worker is currently loading freight onto that same trailer. This worker must wait in queue for the trailer to become available.

3) Transferring Freight

Freight Transferring activity occurs when hub worker use forklift to transfer shipments from one origin trailer to according destination trailer and then go back to the same origin trailer or another origin trailer.

Two travel distributions are used to determine the travel time between doors. The unloaded travel speed is estimated by a uniform distribution UNIF (3.48 feet/second, 4.28 feet/second) with the mean of 3.88 feet/second for the unloaded travel speed. The loaded travel speed is estimated by a uniform distribution UNIF (3.48 feet/second, 4.28 feet/second) with the mean of 3.88 feet/second for the loaded travel speed.

The travel time of hub worker with loaded forklift is calculated by using distance between origin trailer and destination trailer doors divided by the loaded travel speed distribution. Similarly, the travel time of hub worker with empty forklift is calculated by using distance between origin trailer and destination trailer doors divided by the unloaded travel speed distribution. It is assumed that the travel speed distributions are the same for all workers.

Different freight unloading and loading sequence approaches are used in the simulation model to specify the sequence a hub worker unloads and loads shipments. In the trailer-at-a-time approach, a hub worker starts to unload an origin trailer with an arbitrary shipment. When the hub worker finished that shipment at the destination trailer, the worker returns to the origin trailer to transfer another shipment on that origin trailer. The worker repeats this process until all shipments on that origin trailer are finished then this worker goes to work on another origin trailer.

Alternatively, hub workers can use the balance-and-connect approach addressed in Chapter 6 to perform unloading and loading job if all origin trailers are available at the beginning of hub operations. In this approach, the sequences that hub worker unloads and loads freights are determined in advance (from the solutions in Chapter 6). A hub worker starts from first shipment on his job list. Transferring this shipment to the destination trailer, the worker looks at the job list and goes to the origin trailer with the next shipment and transfers that shipment. This process continues until all jobs on the list are complete. To use this approach, it is assumed that all origin trailers are available at the beginning of the hub operations.

The Nearest Neighbor approach is also used in the simulation model. This Nearest Neighbor approach is from Brown (2003). In the Nearest Neighbor approach, all workers can transfer shipments from all origin trailers. When a worker finishes one shipment on a destination trailer, the worker moves to the nearest available origin trailer (without another workers unloading) and transfers a shipment on that origin trailer. The worker repeats this process until all shipments on his job list done. The freight transferring process is summarized in Figure E.3.

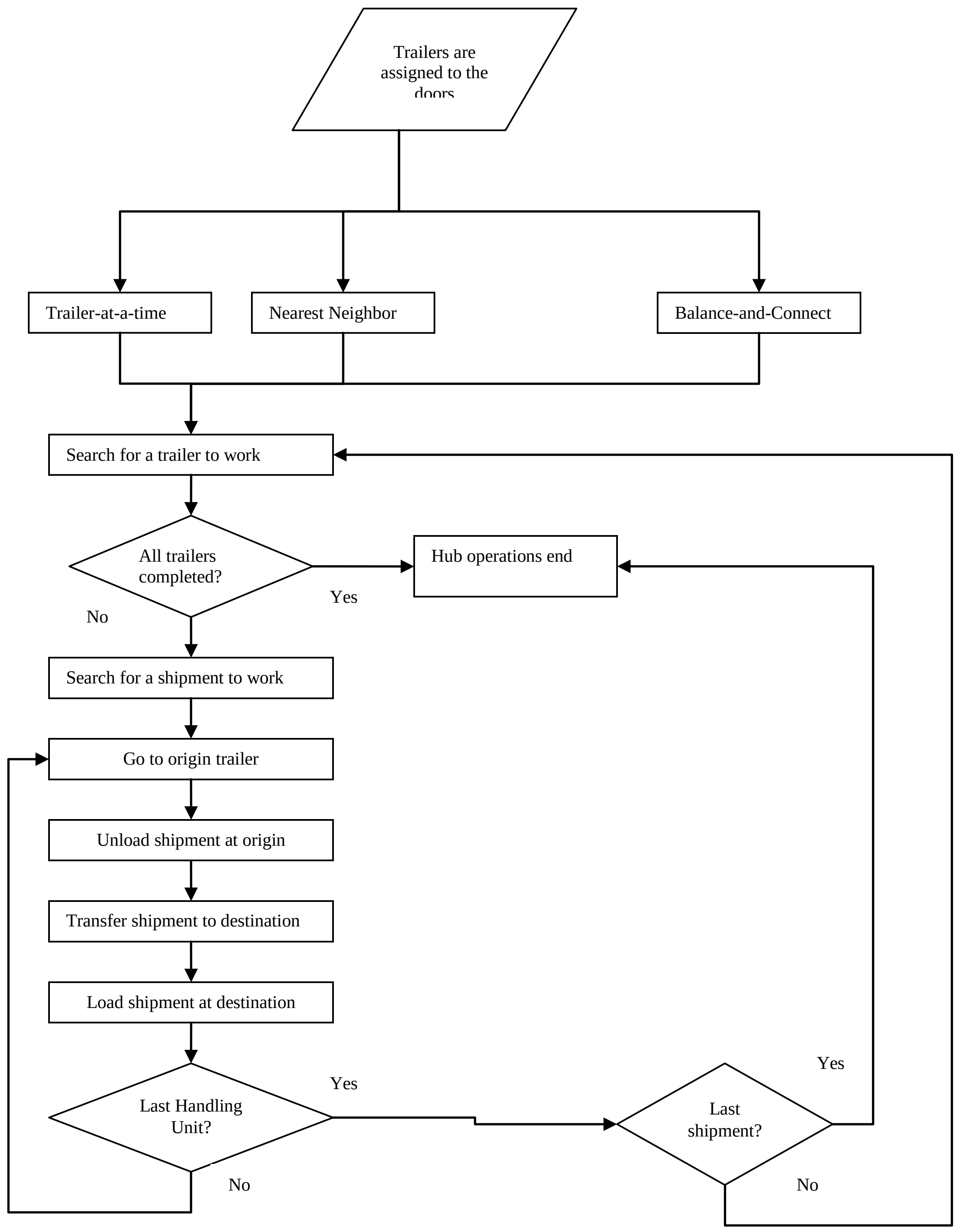


Figure E.3 Freight Transferring Process

E.3 Simulation Model

The simulation model is implemented using Arena (version 7.0). The following entities, resources and sub-models developed in the Arena simulation model are described in this section. The termination condition of simulation model is also discussed in this section. The description of functionalities for entity, resources, etc., in Arena can be found in Kelton (2007).

1) Entities

- Origin and destination trailers. Origin and destination trailers are modeled as entities in the trailers arrival and departure sub-model to trigger the event that trailers are assigned to the dock doors. When the freight transfer process starts, the trailers are enter into the dock doors assigned to them until the origin and destination trailers are finished (in the simulation, origin and destination entities size the door resources until the origin trailer is empty or the destination trailer is full then the sized door resources are released). The origin and destination trailers are also used to trigger the event to collect performance measures statistics and terminate the simulation run.
- Hub workers. Hub workers are modeled as the entities in the freight transfer sub-model to trigger the event that transfer all shipments on the origin trailers. Hub worker entities routed repeatedly from one origin/destination door to another destination/origin door. The hub worker entities size the origin/destination trailer resources as they unload/load shipments on those trailers. When the trailers or shipments assigned to the hub worker are all finished, the hub worker entities trigger the event to collect the performance measures statistics for the hub worker and leave the simulation model.
- Logic entity for reading data. A single logic entity is used in the reading data sub-model to trigger the event to read all data needed for the simulation model.

2) Resources

- Hub Doors. Hub doors are modeled as resources in the trailers arrival sub-model. If a hub door is sized by a trailer, it will not be available until the trailer is done and this hub door resource is released.
- Origin and Destination Trailers. Origin and destination trailers are treated as resources in the freight transfer sub-model. If an origin trailer or a destination trailer is sized by a

hub worker entity, that trailer will not be available until the hub worker entity release the trailer.

3) Sub-models

Three sub-models are used to simulate different hub operations processes:

a. Reading Data Sub-Model

This sub-model will read shipment data (shipment number, origin and destination and handling units for each shipment), trailer data (trailers arrival time, number of shipment on each origin trailer), hub layout data (distance between each pair of doors), hub operations data (number of workers, worker's unloading and loading time, forklift speed), door assignment for each trailer and job sequence for each worker.

Various variables and attributes are created to assign to each trailer and worker. Those variables and attributes are used in the simulation model to help performing functions of trailer arrival and departure process and freight transferring process. The statistics of those variables and attributes will be collected at the end of each simulation. Those statistics will be analyzed to compare different solution approaches what-if scenarios for door assignment problem and freight sequencing problem. The sub-model for reading data sub-model is shown in Figure E.4.

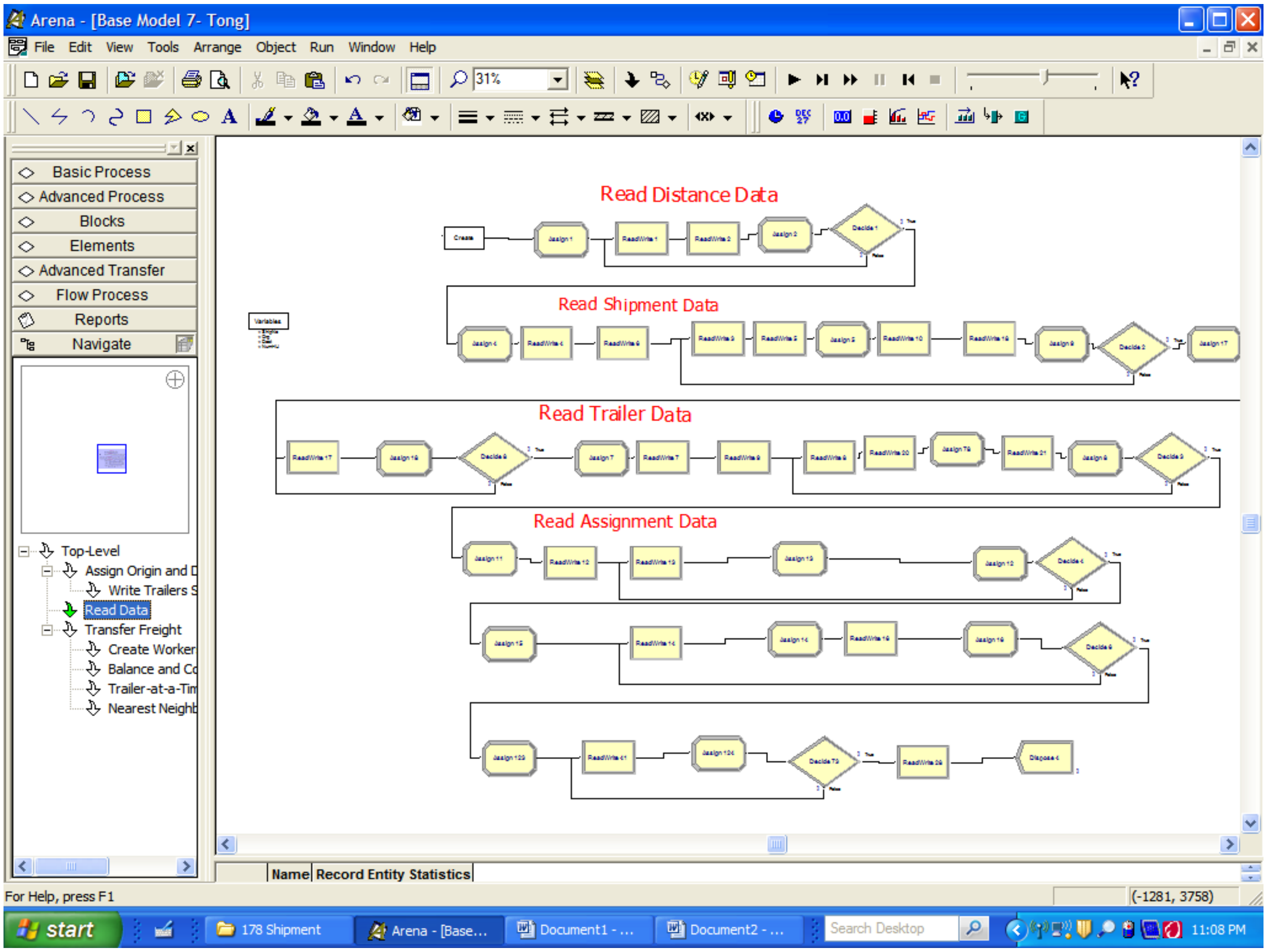


Figure E.4 Reading Data Sub-model

b. Assign Origin and Destination Trailers Sub-Model

This sub-model simulates the trailers arrival and departure process during the hub operations. Origin and destination trailers enter into the simulation model as entities. The origin trailers arrive before the start of hub operations and all destination trailers will be given arrival time 0. All origin trailers arrive after the start of hub operations will be generated as a random number within a certain time range. Each arrived origin trailer and destination trailer seize a hub door according to the door assignment data input.

Once trailers are put into doors they are waiting for hub workers to transfer shipments. The empty origin trailers and full destination trailers will leave their hub doors and those doors will be available for other trailers.

The sub-model for assigning origin and destination trailers sub-model is shown in Figure E.5.

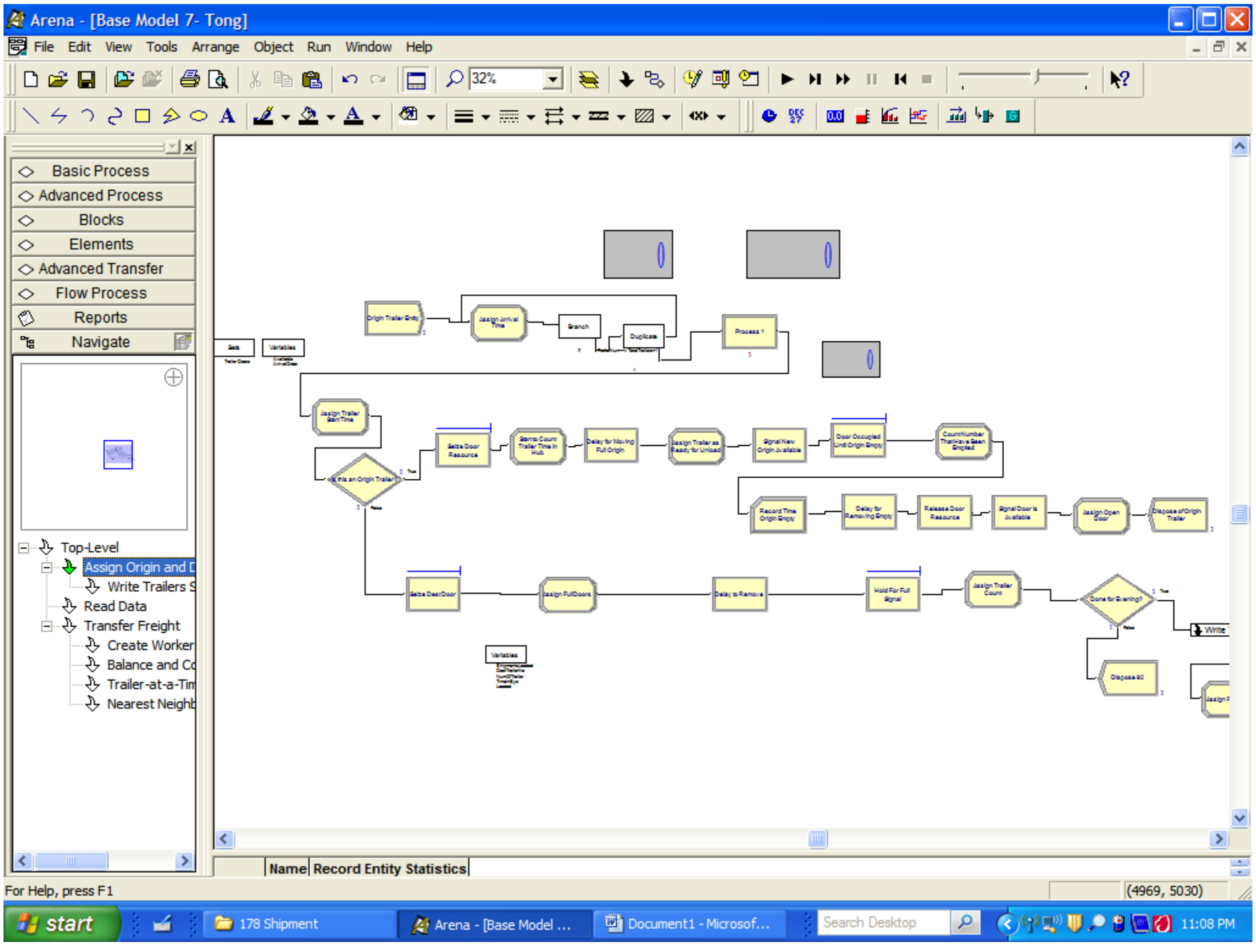


Figure E.5 Assign Origin and Destination Trailers Sub-model

c. Freight Transferring Sub-Model

Hub workers enter into this sub-model as entities to perform freight transferring process. Hub workers first check the number and availability of all origin trailers. If all origin trailers are available and are already set to the hub doors, hub workers will use the job sequence list which from the solution of chapter 6. This job sequence list will indicate the sequence for each shipment that the worker will follow to transfer the shipment. This job sequence also set up a route schedule for the worker when he performs the freight transferring process. In this case in the simulation model workers will follow the set-up route to move one by one shipment from origin trailers to destination doors. In the simulation model this routing task is accomplished by using station and route modules.

On the other hand, if some origin trailers are not available at the beginning of the hub operations, the job sequence list based on the algorithm in chapter 6 is not available for the hub workers. Instead, workers will follow the trailer-at-a-time or nearest neighbor rule to finish origin trailers one by one.

Trailers (and doors) are used as resources in this simulation sub-model. If one origin or destination trailer is seized by a hub worker to unload or load shipment, other hub workers who want to unload or load a shipment at the same trailer must wait until this trailer is available again. The travel speed for the workers follows the same distribution in the simulation model.

Once an origin trailer is empty or a destination trailer is full, a signal indicates the origin trailer or destination trailer may move from the dock door and other waiting trailers (if any) enter the dock door to continue the freight transferring process. The sub-model for origin and destination trailers is shown in the Figure E.6.

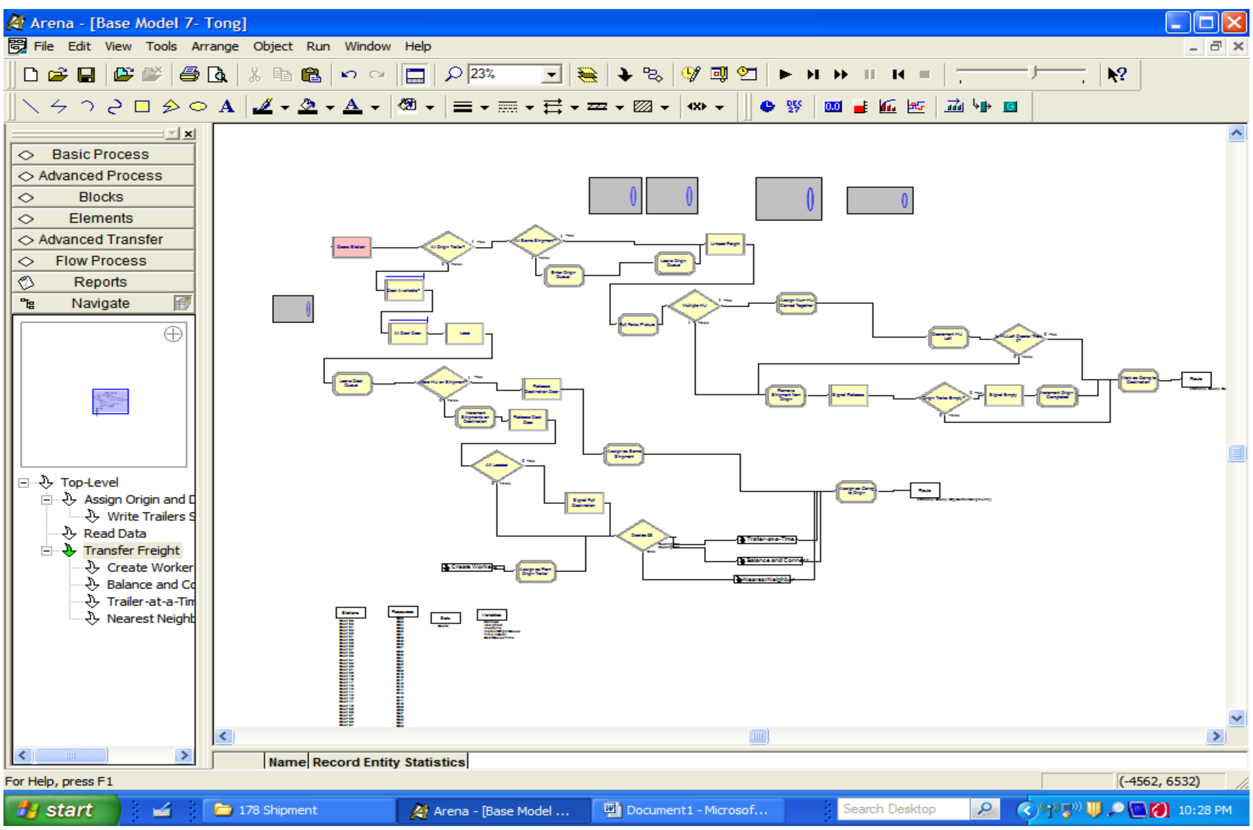


Figure E.6 Transfer Freight Sub-model

4) Termination of Simulation

Simulation can be classified as either terminating simulation or steady-state simulation. Terminating simulation is one in which the simulation starts at a defined state or time and ends when it reaches some other defined state or time. Steady-state simulation, on the other hand, is one in which the steady-state (lone-term average) behavior of the system is being analyzed. By definition, terminating simulation has a fixed starting condition and an event that marks the end of the simulation. In our simulation study, hub operations start at a fix time when all (or most) origin trailers arrive at the hub and hub operations end when all shipments on origin trailers have transferred onto destination trailer. In this sense, hub operations system is modeled as a terminating system in our simulation study. The simulation run is terminated when all shipments are finished and the last destination trailer leaves the hub.

5) Simulation Runs

For each simulation run, a door assignment (trailers assigned to a door number) generated by one of four door assignment approaches read into the simulation model from a spreadsheet. A freight sequencing solution is also set up in a spreadsheet to be read into the simulation model. If the Balanced Trailer-at-a-Time and Trailer-at-a-Time with offloading are used, the sequence number of each shipment is same as the shipment number. If Nearest Neighbor is used, the sequence number of each shipment is determined during the simulation run. If the Assign-First-Route-Second and Balance-and-Connect are used, the worker identifier number and the sequence number for each shipment from the solutions of chapter 6 must be specified in the spreadsheet to be read into the simulation model. All the trailers are assumed to be available at the beginning of the hub operations and the number of hub doors is greater than the number of trailers. The simulation model assigns the trailers to the doors based on the data read in and performs the freight transferring process accordingly. The statistics of performance measures are collected and compared and analyzed.

6) Simulation Performance Measures Definition

The main performance measures evaluated by the simulation model include:

- Total worker time
- Total travel distance

- Transfer time
- Average time a shipment travels at hub
- Total worker travel time
- Bottleneck travel time
- Average workload balance ratio

To define these measures, the following notation is used:

k	index for workers, $k = 1, \ldots, K$
s	index for shipments, $s=1, \ldots, S$
$Tinit_k$	time that worker k begins transferring shipments
$Tend_k$	time that worker k completes transferring shipments
T_k	time required for worker k to complete transferring shipments
	$T_k = Tend_k - Tinit_k$
$Ttravel_k$	time that worker k travels during hub operations
$Dtravel_k$	distance that worker k travels during hub operations

Total worker time is the sum of the time for all the workers to transfer the shipments, such that

$$Total\ worker\ time = \sum_{k=1}^{K} T_k$$

Total travel distance is the distance that all the workers travel to transfer all shipments in the hub operations, such that

$$Total\ travel\ distance = \sum_{k=1}^{K} Dtravel_k$$

The *transfer time* is the time span from the start of the hub operations to the time of last shipment completed. The transfer time is also referred to as the make-span of the hub operations.

$$Transfer\ time = max_k\{Tend_k\} - min_k\{Tinit_k\}$$

Total worker travel time is the sum of the travel time of all the workers, such that

$$Total\ worker\ travel\ time = \sum_{k=1}^{K} Ttravel_k$$

Bottleneck travel time is the travel time for the bottleneck worker, such that

$$Bottleneck\ travel\ time = max_k\{Ttravel_k\}$$

Finally, the *average workload balance ratio* reflects the balance in terms of work time among all workers. Assume T_k is the time required for worker k to complete transferring shipments. Then $\max_k\{T_k\}$ is the time for the bottleneck worker to complete shipment transfers. The *average workload balance* ratio is defined as follows:

$$Average\ workload\ balance\ ratio = \frac{1}{K}\sum_{k=1}^{K}\frac{t_k}{\max\limits_k\{T_k\}}$$

7) Model Verification and Validation

Verification is the process of ensuring that the conceptual model has been transformed into a computer model with accuracy. In process of building the simulation model, many measurements are observed during or after the simulation run to ensure the model accuracy. For example, all shipments are transferred, all trailers are assigned to the right worker, the sequence of unloading and loading for each worker is executed as planned, etc.

Validation is the process of ensuring that the model is sufficiently accurate for real system. In this simulation study, the validation is executed by comparing the results between mathematical models and simulation models. Three key performance measures (bottleneck time, total worker time and total distance) are compared to the results from mathematical model in Chapter 5 and

Chapter 6 and the results from simulation model. The comparisons are summarized in the following tables.

Table E.1 Total Distance Comparison for Data Set 1

	Total Travel Distance (ft.)				
	BTAAT	**NN**	**TAATWO**	**AFRSA**	**BCA**
Simulation Model	79958	77458	79905	77959	70478
Mathematical Model	79072	80103	77821	74465	69040
Percent Difference	1%	3%	3%	4%	2%

Table E.2 Total Worker Time Comparison for Data Set 1

	Total Worker Time (min)				
	BTAAT	**NN**	**TAATWO**	**AFRSA**	**BCA**
Simulation Model	1639	1627	1630	1628	1589
Mathematical Model	1568	1568	1558	1539	1517
Percent Difference	4%	4%	4%	5%	5%

Table E.3 Bottleneck Time Comparison for Data Set 1

	Bottleneck Time (min)				
	BTAAT	**NN**	**TAATWO**	**AFRSA**	**BCA**
Simulation Model	389	355	367	386	269
Mathematical Model	369	347	342	367	253
Percent Difference	5%	2%	7%	5%	6%

Table E.4 Total Distance Comparison for Data Set 2

	Total Travel Distance (ft.)				
	BTAAT	**NN**	**TAATWO**	**AFRSA**	**BCA**
Simulation Model	100804	99806	100901	99557	92037
Mathematical Model	100284	100820	98015	97808	89159
Percent Difference	1%	1%	3%	2%	3%

Table E.5 Total Worker Time Comparison for Data Set 2

	Total Worker Time (min)				
	BTAAT	NN	TAATWO	AFRSA	BCA
Simulation Model	2143	2140	2140	2140	2098
Mathematical Model	2079	2052	2040	2035	1991
Percent Difference	3%	4%	5%	5%	5%

Table E.6 Bottleneck Time Comparison for Data Set 2

	Bottleneck Time (min)				
	BTAAT	NN	TAATWO	AFRSA	BCA
Simulation Model	963	380	390	964	367
Mathematical Model	921	345	343	922	332
Percent Difference	4%	9%	12%	4%	10%

Table E.7 Total Distance Comparison for Data Set 3

	Total Travel Distance (ft.)				
	BTAAT	NN	TAATWO	AFRSA	BCA
Simulation Model	100920	99087	100204	99008	93363
Mathematical Model	99954	101449	99821	97029	89770
Percent Difference	1%	2%	1%	2%	4%

Table E.8 Total Worker Time Comparison for Data Set 3

	Total Worker Time (min)				
	BTAAT	NN	TAATWO	AFRSA	BCA
Simulation Model	2151	2140	2120	2138	2111
Mathematical Model	2064	2055	2053	2031	2001
Percent Difference	4%	4%	3%	5%	5%

Table E.9 Bottleneck Time Comparison for Data Set 3

	Bottleneck Time (min)				
	BTAAT	NN	TAATWO	AFRSA	BCA
Simulation Model	962	382	388	960	367
Mathematical Model	926	361	371	924	334
Percent Difference	4%	5%	4%	4%	9%

The differences between results from the simulation model and mathematical models for the three data sets are all in small range. The total distance has the least difference among all data sets while the total worker time and bottleneck time have larger difference.

E.4 Additional Simulation Experiment 2 Results

In simulation experiment 2, average performance measures for each simulation run are collected and analyzed in Section 7.4. The variances of total worker time and transfer time for each simulation run are also collected and summarized in Table E.10, Table E.11, and Table E.12.

Table E.10 Variances for Total Work Time and Transfer Time in Experiment 2 for Data Set 1

Simulation Run	Door Assignment	Freight Sequencing	Total Worker Time Variance (min)	Transfer Time Variance (min)
1	SPW	BTAAT	10.69	2.41
2	SPW	NN	10.61	2.79
3	SPW	TAATWO	12.39	3.11
4	SPW	AFRSA	10.67	2.41
5	SPW	BCA	10.48	2.14
6	DPW	BTAAT	9.88	3.12
7	DPW	NN	13.07	2.26
8	DPW	TAATWO	13.13	2.60
9	DPW	AFRSA	9.86	2.33
10	DPW	BCA	9.71	1.68
11	DGA	BTAAT	13.12	2.34
12	DGA	NN	9.82	2.63
13	DGA	TAATWO	9.80	2.22
14	DGA	AFRSA	9.83	3.10
15	DGA	BCA	9.63	1.64
16	DSA	BTAAT	13.11	2.33
17	DSA	NN	9.76	2.49
18	DSA	TAATWO	11.41	2.20
19	DSA	AFRSA	9.77	2.70
20	DSA	BCA	9.53	1.61

Table E.11 Variances for Total Work Time and Transfer Time in Experiment 2 for Data Set 2

Simulation Run	Door Assignment	Freight Sequencing	Total Worker Time Variance (min)	Transfer Time Variance (min)
1	SPW	BTAAT	12.33	5.76
2	SPW	NN	14.75	3.23
3	SPW	TAATWO	17.03	3.90
4	SPW	AFRSA	12.33	5.76
5	SPW	BCA	11.78	2.69
6	DPW	BTAAT	17.44	5.96
7	DPW	NN	10.86	3.33
8	DPW	TAATWO	10.84	2.53
9	DPW	AFRSA	10.88	5.96
10	DPW	BCA	12.82	1.56
11	DGA	BTAAT	10.73	5.78
12	DGA	NN	10.81	2.48
13	DGA	TAATWO	15.11	2.52
14	DGA	AFRSA	10.81	7.79
15	DGA	BCA	12.95	1.93
16	DSA	BTAAT	10.72	3.85
17	DSA	NN	12.84	2.28
18	DSA	TAATWO	10.70	1.95
19	DSA	AFRSA	12.84	5.78
20	DSA	BCA	12.59	2.20

Table E.12 Variances for Total Work Time and Transfer Time in Experiment 2 for Data Set 3

Simulation Run	Door Assignment	Freight Sequencing	Total Worker Time Variance (min)	Transfer Time Variance(min)
1	SPW	BTAAT	12.55	4.68
2	SPW	NN	17.33	2.17
3	SPW	TAATWO	12.53	1.78
4	SPW	AFRSA	10.01	3.48
5	SPW	BCA	12.37	1.80
6	DPW	BTAAT	17.54	1.99
7	DPW	NN	10.86	1.68
8	DPW	TAATWO	10.85	2.10
9	DPW	AFRSA	13.08	2.97
10	DPW	BCA	10.85	1.54
11	DGA	BTAAT	17.19	4.81
12	DGA	NN	10.77	1.67
13	DGA	TAATWO	17.24	1.68
14	DGA	AFRSA	10.85	5.83
15	DGA	BCA	6.53	1.55

16	DSA	BTAAT	10.76	1.92
17	DSA	NN	4.28	1.53
18	DSA	TAATWO	10.60	0.78
19	DSA	AFRSA	8.55	2.88
20	DSA	BCA	12.67	1.47

E.5 Simulation Experiment 1 Results for Data Set 2 and Data Set 3

In Chapter 7, the impact of door assignment to the performance measures of hub operations is illustrated by comparing total distance, total worker time and transfer time and average workload ratio from the dynamic approaches (DPW, DGA and DSA) to the same performance measures from semi-permanent approach (SPW), using the simulation results from data set 1. In this section, the comparisons using the simulation results from data set 2 and data set 3 are included. The comparisons also support the conclusions from Chapter 7.

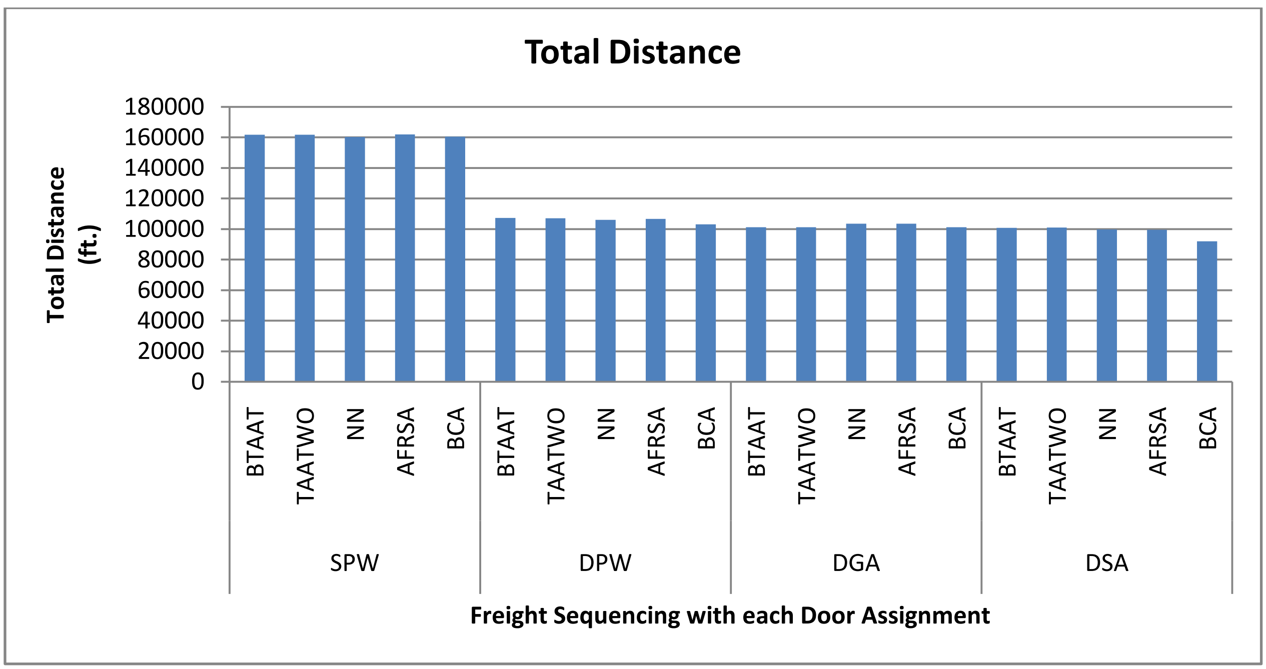


Figure E.7 Total Distance Comparison from Data Set 2 for Simulation Experiment 2

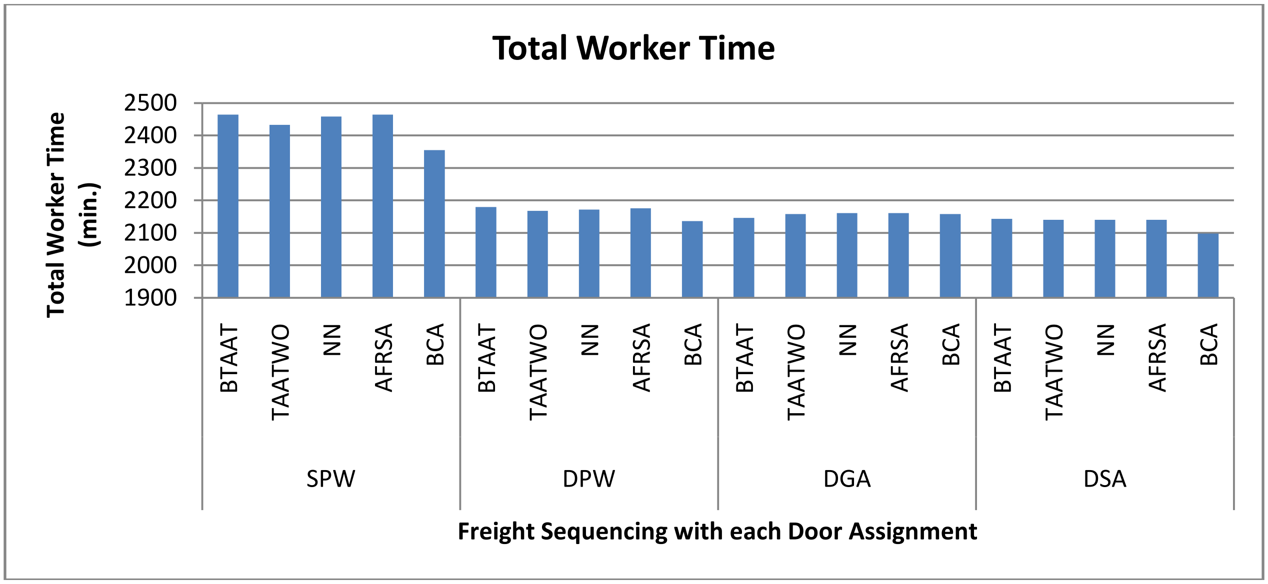


Figure E.8 Total Worker Time Comparison from Data Set 2 for Simulation Experiment 2

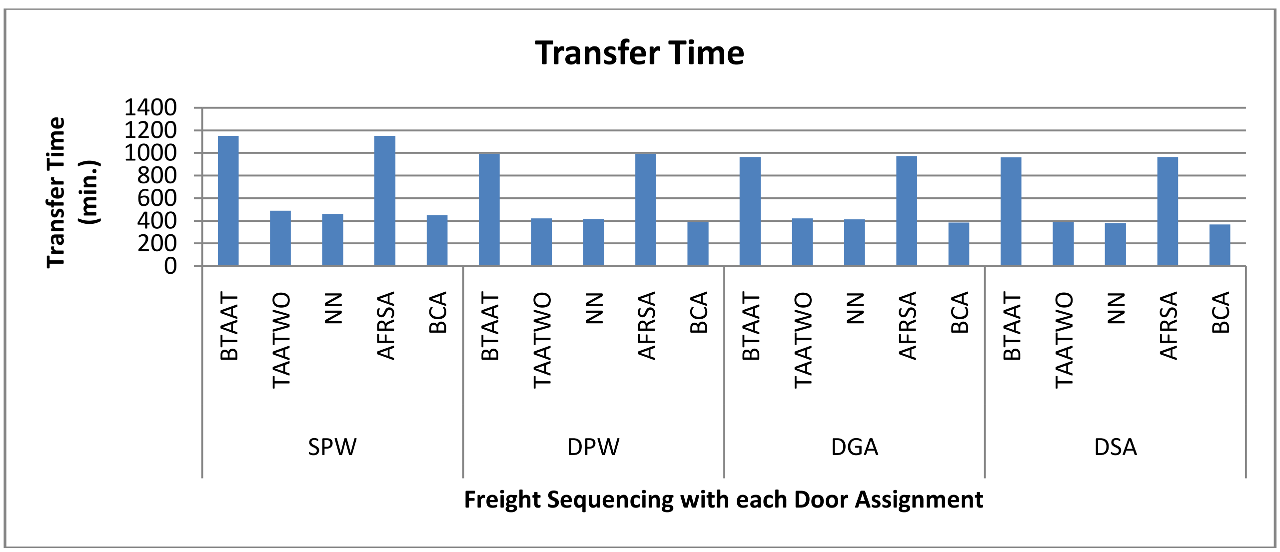


Figure E.9 Total Transfer Time Comparison from Data Set 2 for Simulation Experiment 2

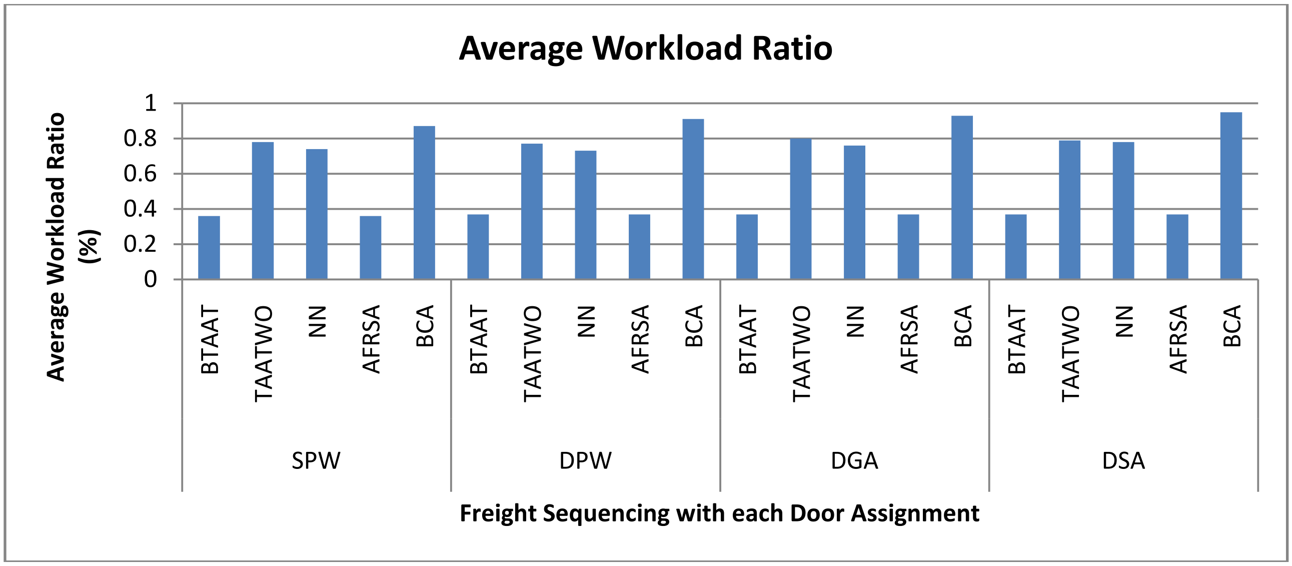


Figure E.10 Average Workload Ratio from Data Set 2 for Simulation Experiment 2

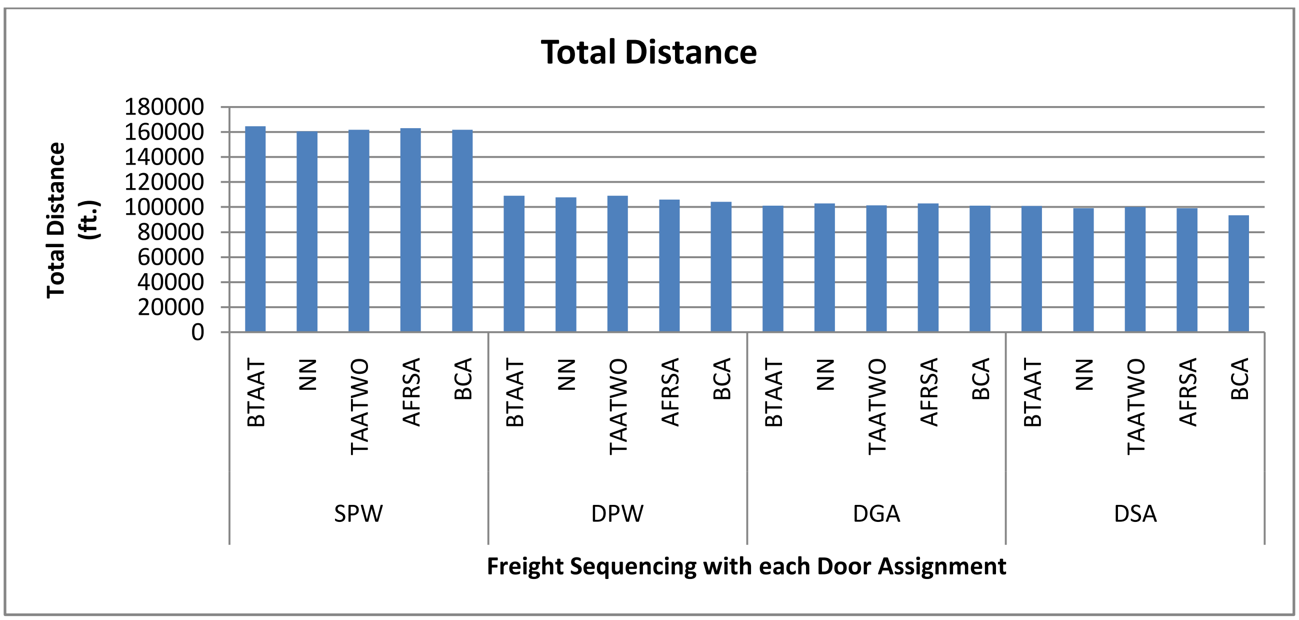


Figure E.11 Total Distance Comparison from Data Set 3 for Simulation Experiment 2

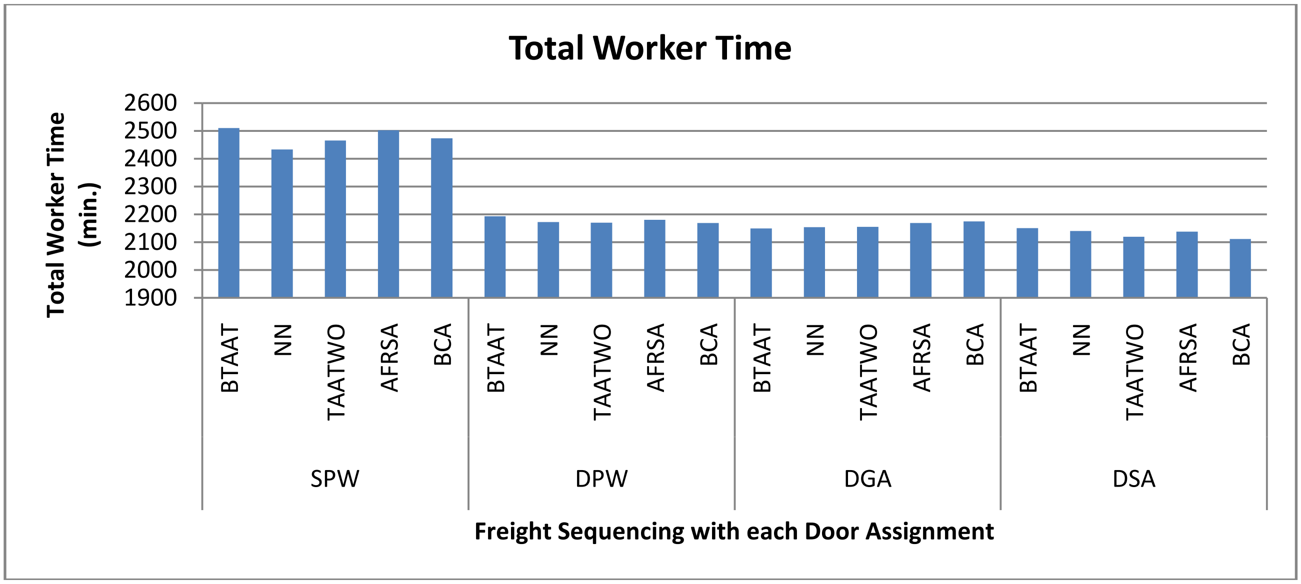


Figure E.12 Total Worker Time Comparison from Data Set 3 for Simulation Experiment 2

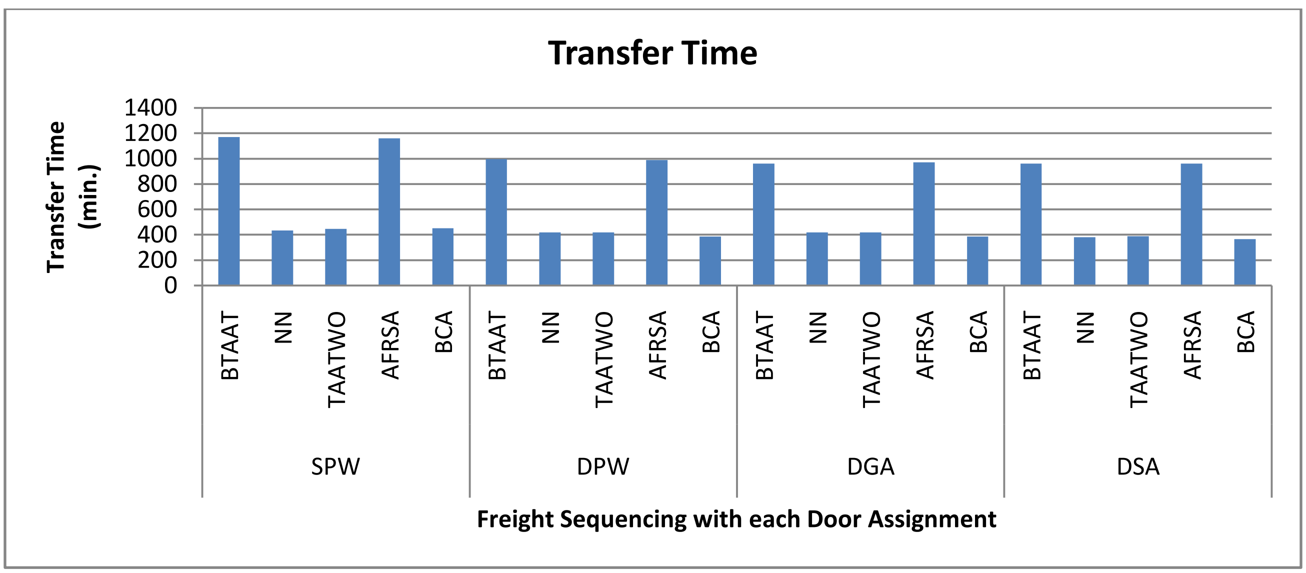


Figure E.13 Total Transfer Time Comparison from Data Set 3 for Simulation Experiment 2

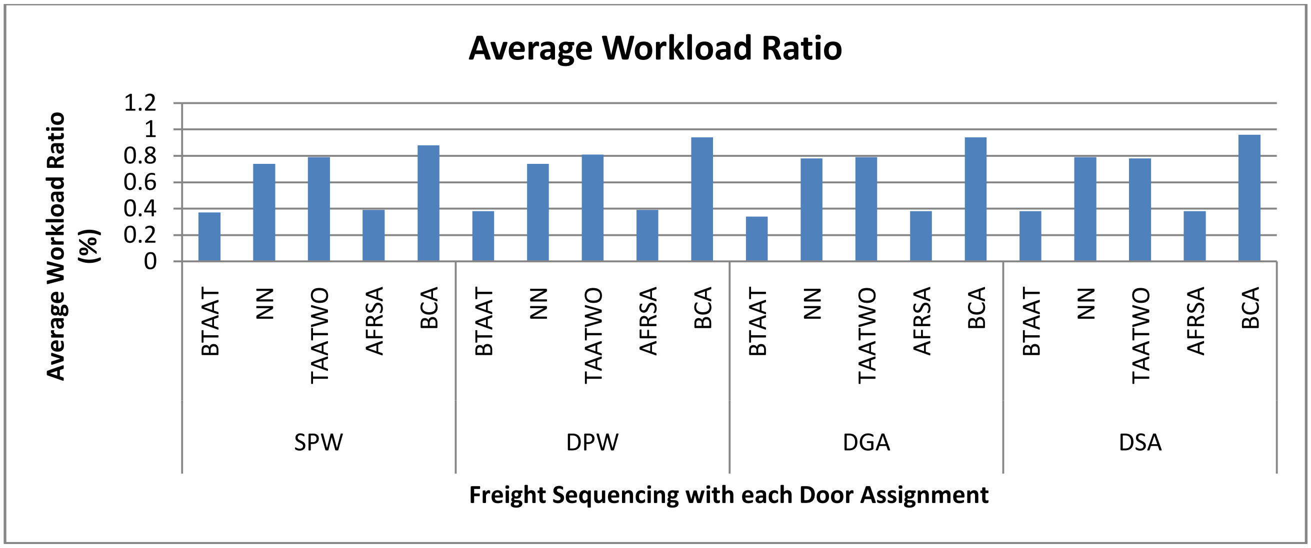


Figure E.14 Average Workload Ratio from Data Set 3 for Simulation Experiment 2