To the University Council:

The Thesis Committee for Kaifeng Ji certifies that this is the approved version of the following thesis:

TRUCK SCHEDULING PROBLEM AT A CROSS-DOCKING FACILITY

Martin E. Lipinski, Ph.D.
Major Professor

Mihalis M. Golias, Ph.D.

Stephanie Ivey, Ph.D.

Accepted for the Graduate Council:

Karen D. Weddle-West, Ph.D.
Vice Provost for Graduate Programs

TRUCK SCHEDULING PROBLEM AT A CROSS-DOCKING FACILITY

by
Kaifeng Ji

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Civil Engineering

The University of Memphis

April, 2010

ABSTRACT

Ji, Kaifeng. M.S. The University of Memphis. May 2010. Truck Scheduling Problem at a Cross-docking Facility. Major Professor: Martin E. Lipinski, Ph.D.

As a relatively new concept of warehousing with limited storage time to maximize the throughput, cross-docking plays a significant role in an increasing number of companies' warehousing policies in today's customer driven economy. Problems relating to cross-dock facilities can be categorized into two groups: a) problems that consider the facility as a node within a larger transportation network; and b) problems that focus on the operations of the facility (Boysen and Fliedner 2010). In this thesis, the latter type of problem is considered, and two truck scheduling models with handling time as a variable are proposed. In the first model, Just-in-time scheduling with a time window to evaluate the operation process is used. A memetic algorithm and a number of computational examples to show the advantage are also developed. In the second model, the total service time of trucks and the temporary storage time inside the facility are minimized using the function of handling time. A multi-objective memetic algorithm is proposed to solve this model. Finally, the limits of current models and the direction of future works are described.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

In today's customer driven economy, moving products quickly and cost efficiently provides a distinctive comparative advantage for companies. To this effect, an increasing number of companies are finding that cross-docking operations can play an integral part of their distribution model, partially replacing or complementing existing warehousing policies.

Cross-docking is a special warehousing policy moving goods directly from inbound trucks (ITs) to outbound trucks (OTs) without storage or with just temporary storage. In a typical logistics distribution network, products are sent to a warehousing facility for storing, retrieving, sorting and reconsolidating (Sunil and Meindl, 2002; van den Berg and Zijm, 1999; Zäpfel and Wasner, 2006). Products are subsequently sent out to retailers upon requests (Baker, 2008). However, as inventory costs represent one of the major costs in a supply chain, cross-docking become an attractive alternative to warehousing. Using cross-docking facilities in the supply chain, companies will benefit from moving products fast and saving inventory costs.

In contrast to traditional warehouses, the operation process inside a cross-docking facility includes receiving goods from inbound trucks (IT) at inbound doors (ID), and sorting and shipping goods to outbound trucks (OT) at outbound doors (OD). In cross docking, the inventory cost is minimized since it takes little or limited storage, thus making it an attractive alternative to warehousing. However, the labor cost associated with loading/unloading, sorting, and transferring is increased, and effort must be made to develop efficient scheduling policies to address this. Another important issue to consider in cross-docking is increasing efficiency in transshipment time to avoid any early or late departure as either can lead to increase cost. The objective of this study is to consider the truck scheduling problem which encompasses the scheduling of inbound trucks (ITs) and outbound trucks (OTs) to inbound doors (IDs) and outbound doors (ODs) to obtain efficient transshipment time, reduce temporary storage inside the facility and ensure on-time deliveries.

In order to solve the truck scheduling problem at a cross-docking facility to obtain an economically efficient transshipment operation, it is necessary to optimize both ITs-to-IDs and OTs-to-ODs assignments. One approach is to treat the problem using a machine scheduling method. In machine scheduling modeling, IDs and ODs are considered as two machines serving their own jobs (ITs and OTs). Usually, mathematical optimization models are used to solve the machine scheduling problem by optimizing the location and the order of jobs to be served, which is identical to cross-docking truck-to-door assignment.

Using machine scheduling, several models have been developed to solve the truck scheduling problem. However, in the current research and mathematical models on this topic, handling time of each truck inside the cross-docking facility is always estimated as a constant input or normalized data instead of a variable depending on truck-to-door assignment. However, in real life operations, the handling time of each inbound truck depends on its location, assignments of outbound trucks it serves and the number of forklifts sent to serve it. This is a limitation of truck scheduling models.

In order to improve the current research on the truck scheduling problem, this thesis is focused on developing a formulation of handling time as a variable of truck-to-door assignment and scheduling the ITs and OTs to available IDs and ODs. Two models are presented here with different objectives to simulate cross-docking operations for Just-in-time scheduling and zero inventory cross-docking policies.

In cross-docking operations, trucks usually have departure time requests for operators to meet. In this situation, ITs and OTs need to be scheduled just in time to meet this requirement and the service time and costs also need to be reduced. In the first model presented, the Just-in-time scheduling policy which is very similar to real-world operations is used to schedule ITs and OTs to IDs and ODs. Compared to previous research (Li, et al. 2004; Alvarez-Perez, et al. 2009), departure requests are introduced here in the form of a time window, where the facility operator is penalized if that departure time window is not met (as opposed to the point in time departure in (Li, et al. 2004; Alvarez-Perez, et al. 2009)). Minimization of both early and tardy departures is consistent with a just-in-time (JIT) philosophy, where both earliness and tardiness are discouraged. An ideal schedule is therefore one where all trucks depart the cross dock facility within the requested time window. Minimizing the total delayed or early departures will affect the productivity of the facility (i.e. total throughput). This aspect is considered in the model presented here by including an additional term minimizing the total service time for all the trucks. Also, truck handling time is assumed to be not only dependent on the amount of cargo (un)loaded but also on the truck-door assignment. It is formulated as a function of the distance the forklifts will travel carrying cargo from the ITs to the OTs. Thus using this model formulation in cross-docking, reasonable handling time estimation would help with scheduling ITs and OTs to IDs and ODs more close to real operation situations. This is the first time in the published literature that such a truck scheduling policy under these assumptions has been presented and represents the key contribution of this work. The mathematical formulation is a single-objective problem that is nonlinear and NP-Complete, even in the simple case of one ID and one OD. To tackle this issue and solve the resulting problem with reasonable computational effort, a memetic algorithm is developed and the scheduling policy and solution algorithm are evaluated through a number of numerical examples.

Another issue in the truck scheduling problem, as mentioned before, is that temporary storage inside the cross-docking facility might occur when trucks are assigned to specific doors. If this happens, congestion affects the truck-to-door assignment result and cost is increased by maintaining the temporary storage. Considering the congestion situation and temporary storage, a second bi-objective model is also proposed to assign incoming and outgoing trucks to the inbound and outbound doors with two objectives: a) minimization of the total service time for all the trucks served at the facility, and b) minimization of the total storage time of the commodities transferred from the incoming to the outgoing trucks. In this model, it is also assumed that truck handling times depend on both the amount of cargo (un)loaded and the truck-to-door assignment. Using this handling time formulation, every detailed time period inside the facility can be estimated. Thus under this bi-objective model, storage time and service time will both be minimized while doing truck-to-door assignment.

In conclusion, two mathematical models are presented in this thesis to solve the truck scheduling problem with the objectives of minimizing total service time, minimizing total cost from tardy and early departures and minimizing total storage time. In both models formulations, handling time inside the facility is treated as a variable function of the door assignment of both sets of trucks.

The rest of the thesis is organized as follows: Section 2 presents a comprehensive literature review describing warehousing operations, cross-docking considerations, and models dealing with the operations, scheduling and optimization problems at a cross dock facility. Section 3 describes the first model with the objective of minimizing the total cost under a Just-In-Time scheduling policy. Section 4 provides a computational example of the first model using a memetic algorithm to solve the resulting problem. Section 5 describes the second model which is a bi-objective model with two objectives: a) the minimization of the total service and b) the minimization of the total storage time at a cross-docking facility. Section 6 proposes a solution algorithm for the second model. This thesis concludes with suggestions for the direction of future research work.

2. LITERATURE REVIEW

2.1 Warehouse Distribution

Before we have warehouse, the shipping from suppliers to customers is direct distribution:

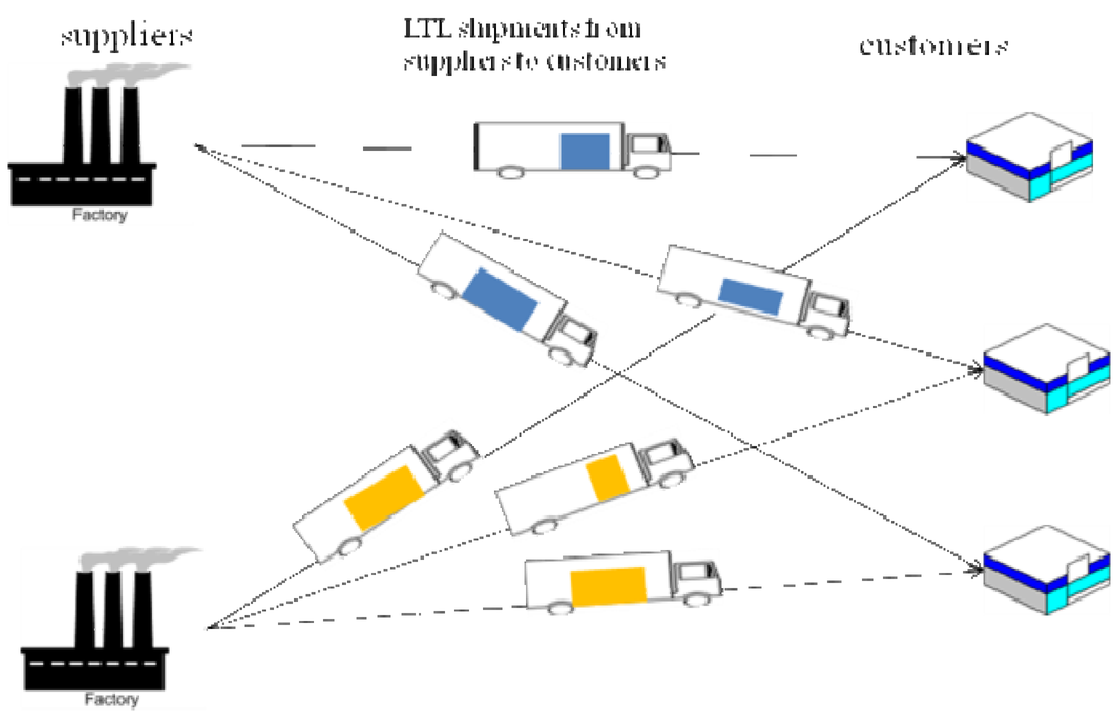

Figure 2.1 Direct Distribution

In the direct distribution, suppliers need to send trucks to every customer they have and mostly in less-than-truckload (LTL). At the same time, customers such as retailers need to order multiple trucks from different suppliers for just one order. This is a waste of resource and not economic for both suppliers and customers. In this case, warehouse appeared to build an intermediate between suppliers and customers.

Generally speaking, a warehouse is a commercial building for storage of goods. Warehouses are used by manufacturers, importers, exporters, wholesalers, transport businesses, customs, etc[1]. They usually have loading docks to load and unload goods from trucks. Using warehouses, companies can consolidate product, deal with the sudden order changes, and thus better match customers' demand.

The process of warehouse operations can be simply defined as three processes (Figure 2.2):

1 Inbound processes

- Receive
- Put-away

2 Storage processes

- Consolidate

3 Outbound processes

- Order picking
- Checking, packing, shipping

---

[1] wikipedia, "warehouse", Wikipedia, <http://en.wikipedia.org/wiki/Warehouse>

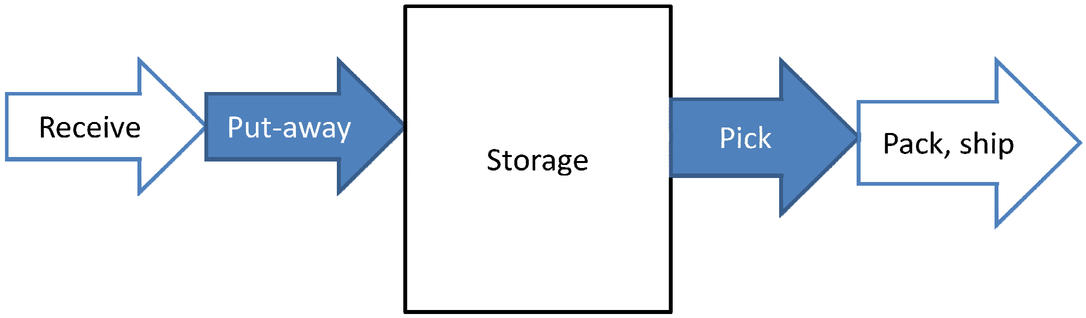

Figure 2.2 Warehouse Operation Process

After having warehouse in between, the supply chain becomes what we called warehouse distribution (Figure 2.3):

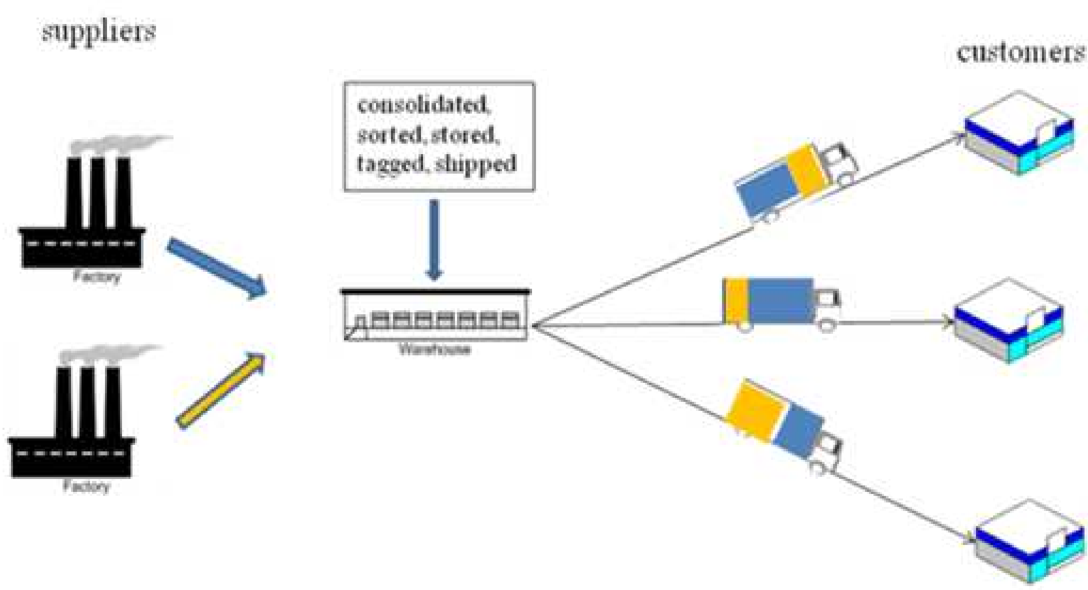

Figure 2.3 Warehouse Distribution

We can see from Figure 2.3 that as an intermediate, warehouse provides storage for suppliers and serves customers' orders by consolidating, pick-up, packing and shipping. Full truckload shipping is possible in this case. Costs on storage and shipping are saved efficiently.

2.2 Types of Warehouses

Based on customer types, warehouse can be categorized into four different types (Rouwenhorst, et al. 2000) :

1. ***Factory Warehouse---Interfaces production with wholesalers***: The factory warehouse has a small number of large orders and advance information about order composition. In factory warehouses, since they are usually operated by the producers, they only have certain kinds of goods and they focus their eyes on the cost and order accuracy.
2. ***Retail Distribution Center---Serves a number of captive retail units***: This kind of warehouse has to provide a large number of orders involving different kinds of retail units. As a result, accuracy information about order composition and the picking process is required. The picking operation inside the warehouse is more complicated because of huge amount orders of different items. The labor cost is a main thing to consider.
3. ***Catalog Retailer---A warehouse filling orders from catalog sales***: This type of facility usually serves a large number of small (frequently single-line) orders. The picking process according to the composition of orders is usually different. And this type of warehouse has a requirement for response time.
4. ***3PL (Third-Party-logistics) Warehouse---A stock room providing raw material and/or work-in-process to manufacturing operations.*** This warehouse type usually

has stringent time requirements, which means fast responses are necessary (e.g., 30 min) for the large amount of small orders. The primary focus is on the response time and the accuracy, while reducing costs.

According to today's customer driven economy, moving products quickly, efficiently, and cost effectively would offer a distinctive comparative advantage to companies. This is the developing trend for warehousing. And we have to make our warehousing process faster and faster to meet the requirement of customers. To this effect, a new concept of warehousing with limited storage but effective throughput rate becomes a popular warehousing policy for many producing, retailer and as well as 3PL companies.

2.3 The Idea of Cross-docking

Cross-docking is a practice in logistics of unloading materials from an incoming truck or rail car and loading these materials directly into outbound trucks, trailers, or rail cars, with little or no storage in between.[2]

As inventory costs represent one of the main cost in a supply chain, cross-docking becomes an attractive alternative to warehousing. Cross-docking is a material handling operation, where products move quickly and directly from inbound trucks (ITs) to outbound trucks (OTs), after being resorted or consolidated with limited storage needs, normally not exceeding 24 hours (Saxena 2007; Laumar 2008). These types of facilities are generally used in "hub-and-spoke" arrangements, where (de)consolidation of cargo

[2]Wikipedia, "cross_docking", Wikipedia, <http://en.wikipedia.org/wiki/ Cross_docking >

occurs as in the case of transshipment, with products delivered to customers in truckloads (TL) (Figure 2.4).

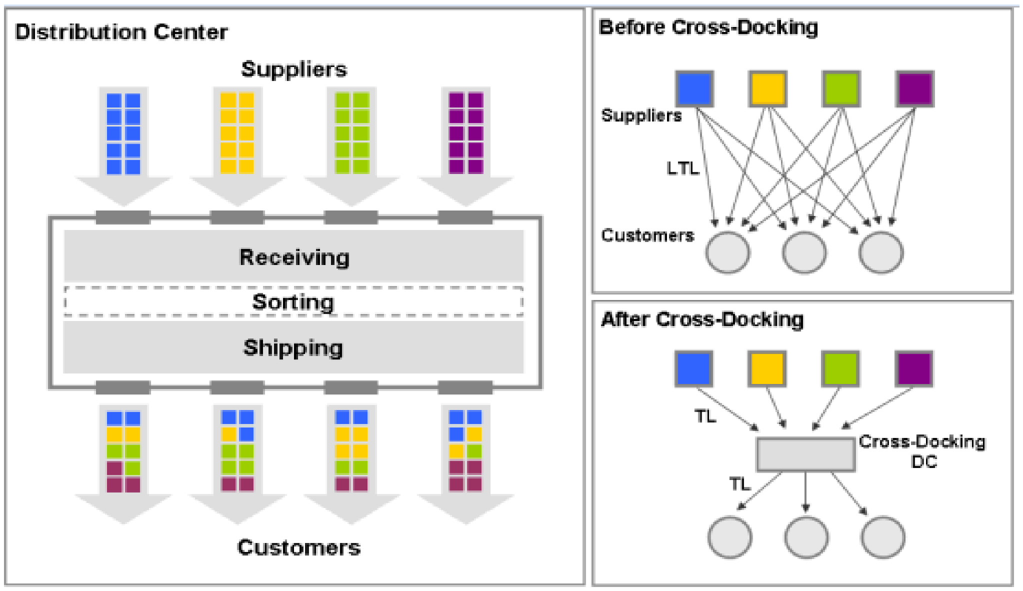

Figure 2.4 Cross-docking Operation Process[3]

As a result, the supply chain is connected directly from point of origin (supplier) to point of sale (retailer), the commodity is moving faster, the inventory costs, handling costs, and operating costs are reduced and customer's just-in-time demands are better matched.

Since first pioneered by the Wal-Mart Corporation -where about 85% of its commodities are delivered through cross dock facilities- companies are increasingly starting to adopt cross dock operations. A survey of 547 industry professionals, carried out by Saddle Creek, showed that 52% of the respondents used cross dock and 13% plan to do so within the next one to two years (Saxena 2007; Laumar 2008; Creek 2008).

[3] http://people.hofstra.edu/geotrans/eng/ch5en/conc5en/img/crossdocking.gif

2.4 Cross-docking Facilities Layout

Cross-docking facilities may be very different in shape. For example, L, I, T are the most common shapes for docks, but you can also find some unusual shapes like U, E, H (Figure 2.5).

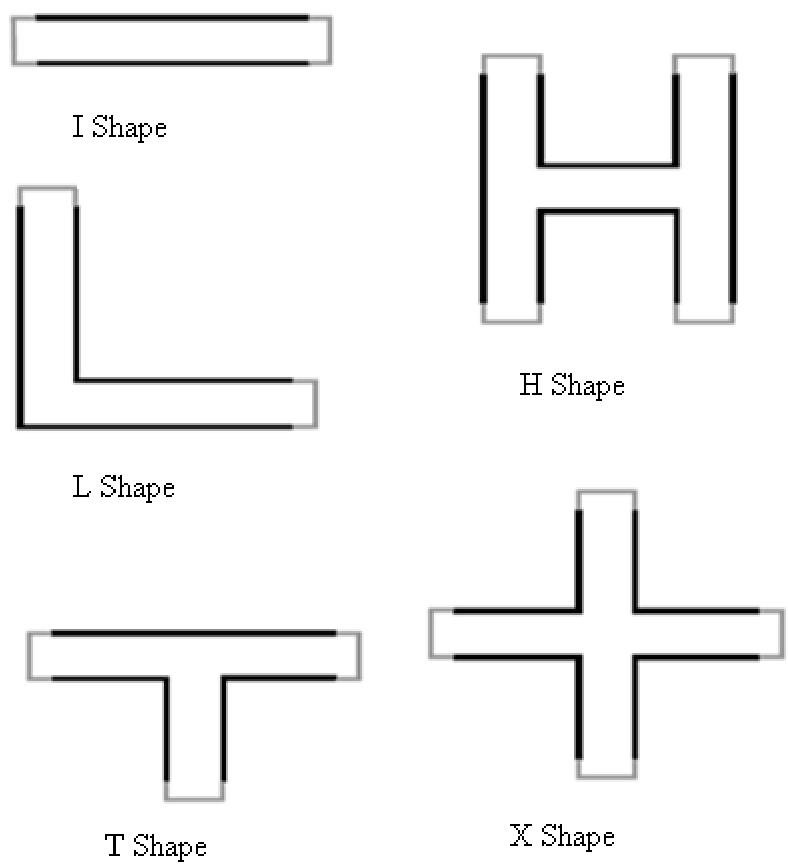

Figure 2.5 Shapes of Crossdocks (Bartholdi and Gue 2004)

Bartholdi and Gue 2004 provides a comprehensive discussion on the connection between crossdock shapes and freight travel distance inside (which represents labor cost), and demonstrates that the best shape for a crossdock depends on the size of crossdocks and the pattern of freight flows inside.

In fact, this is easy to understand. As mentioned before, with limited or no storage process inside, the cross-docking facilities operating cost would be just the labor cost of doing handling and operations. Labor cost depends in large part on how trailers are assigned to doors around the dock; that is, on its layout (Bartholdi and Gue 2000). While moving goods inside the facilities where the ODs are located, the distance from the IDs and the congestion of the commodity flow will affect the travel distance of labor, which represents the cost. Thus, based on the size and other requirements of the cross-docking facilities, determining a reasonable layout and optimizing truck-to-door assignment are the most efficient ways to reduce labor cost and improve the throughput rate.

2.5 Operational and Planning Problems at a Cross-docking Facility

According to the operation process of cross-docking, there are several operational and planning problems to be solved with a cross-docking facility. As mentioned in subsection 2.4, it is important to decide the shape and the layout of the crossdock first, where the IDs and ODs are located. Next it is necessary to determine a truck scheduling policy to assign ITs and OTs to IDs and ODs, for example, first-come-first-serve. Once trucks are assigned to doors, a forklift assignment must take place to determine how many forklifts should be sent to serve one specific truck considering the whole operation process and the congestion which may occur. All of these problems are related to the operational cost of the cross-docking facility. Thus an appropriate way to solve the problems and reduce the total cost is desired.

2.6 Truck Scheduling Problem

The truck scheduling problem decides the assignment of truck-to-door which is the most important factor affecting the whole operation process of the cross-docking facility. In cross-docking facilities, there are IDs and ODs on both sides. ITs and OTs usually arrive at the door randomly. In addition, one OT may receive goods which come from several ITs. Once you have the ITs and OTs' location, the handling time and travel distance is fixed. This means that if different truck-to-door assignments are tried, a process time (or cost) result for each assignment can be determined so that an optimal design can be identified. Thus, mathematical optimization model is a very appropriate way to address the truck scheduling problem.

2.7 Problem Classification

Problems relating to cross dock facilities can be categorized into two large groups: a) problems that consider the facility as a node within a larger transportation network, and b) problems that focus on the operations of the facility (inbound doors, staging, and outbound doors). The former problems (Donald, et al. 1999; Sung and Song 2003; Dobrusky 2003; Lee, et al. 2006; Wen, et al. 2008) include: a) the routing of vehicles from/to the cross dock facility, b) the location and the demand allocation to the cross dock facility, and c) the design of the supply chain network given the cross dock facility. The latter problems (Miao, et al. 2006; Song and Chen 2007; Wang et al. 2008; Bozer and Carlo 2008; Yu and Egbelu 2008, Boysen, et al. 2008) include: a) optimization of operations at the inbound doors (IDs), b) optimization of operations at the outbound doors (ODs), c) optimization of operations within the storage area of the cross dock facility. ID operations consist of the assignment of a time slot; door; unloading cargo from the ITs; recording of data on incoming products and their characteristics; and assignment of temporary storage location, if needed. OD operations consist of the assignment of a time slot and door; loading cargo to the OTs; generation of manifests; and recording of information on shipment and vehicle. Operations within the temporary storage area consist of the allocation of temporary storage space to the incoming cargo; deconsolidation of cargo; planning of packing and consolidation of materials; locator systems; etc. Cargo arriving at the cross-dock facility may be loaded directly onto an OT (one-touch complexity); staged on the dock and then loaded onto an OT (two-touch complexity); or staged on the dock, reconfigured and then loaded on an OT (multiple-touch complexity). Depending on the complexity of the cross-dock facility (one-touch, two-touch, multi-touch), optimizing the different operations can become rather tedious. As the planning of cross-dock facilities includes the scheduling of inbound and outbound transportation, which makes the problem more dynamic than mere warehousing operations, improvements in this area have appeared only recently (Laumar 2008). One of the most important functions in a cross-dock environment is the determination of those docks to which incoming and outgoing trucks should be assigned.

2.8 Literature Review

Many different approaches have been considered on this truck-to-door assignment in cross docking facilities. Most focus on developing models to solve truck scheduling problem with the objective being to minimize the cost and the service time. The difficulty with scheduling models is finding an appropriate variable to represent the cost and time.

As a basic concern in cross docking facilities, minimizing cost is the primary goal to achieve. Tsui and Chang 1990 and Tsui and Chang 1992 provided a model which can be used to minimize the distance traveled by forklifts. In their model, each trailer is assigned to a forklift driver and each shipping door is assigned to only one destination. In this case, the shorter distance forklifts traveled, the less it would cost. However, this model is built under the condition that the number of doors exceeds the number of trucks. The information of OTs-ODs assignment is required to do ITs-IDs assignment.

In Lim, et al. 2006, this problem was extended by specifying truck arrival times and departure times when the number of trucks is greater than the capacity of a crossdock. An integer programming model was used with the objective to minimize the operational cost of the cargo shipment and the number of unfulfilled shipments. By giving each truck several parameters such as operational time, operational cost per unit time and penalty cost per unit cargo, the two objectives are combined into one term: the total cost, a sum of the total dock operational cost and the total penalty cost for the unfulfilled shipments (which represents the maximum throughput rate). However, in this model, operational time of each truck is not considered as a variable, thus limiting its ability to realistically model operations.

Bartholdi and Gue 2000 defined the cost in cross docking facilities as consisting of three parts: Costs for drivers and vehicles making pickups and deliveries, line haul costs and handling costs. In their model, weighted distance is used to represent cost. In addition, congestion is considered as an impact on the handling process and man-hours are used to estimate cost. As a result of congestion, they considered both worker travel time and worker waiting time as input data.

Because of different cross docking operations, different policies have been applied to the cross docking truck scheduling problem due to the specific characteristics of the issue. In initial research, the First-come-first-serve policy was a widely used scheduling policy in models until Gue 1999 developed the Look-ahead scheduling policy to reduce labor costs. For example, a trailer will get priority if its goods require rapid turnaround. Wang, et al. 2008 followed by proposing the Leave-early algorithm which focused on how to assign a truck from a waiting line to be unloaded so that one or more outgoing trucks leave early. However, in these models, handling time is still not considered as a variable of truck-to-door assignment.

Similarly, Boysen 2009 provided a scheduling model at zero-inventory cross docking terminals. There is no storage process in this case. As a result, outbound trucks are scheduled to leave the facility at the earliest point in time as possible. This model has a new idea of minimizing the total storage inside the cross-docking facility which will affect the handling process and the departure time requests of the trucks. However, handling time is an input constant here in the model.

As a detailed scheduling model, Yu and Egbelu 2006's model has always been considered to be very important. This approach addressed the concept of throughput rate which needs to be maximized and developed an objective function to minimize total operation time of the cross docking process. This model is a very detailed scheduling problem model which is similar to that proposed for this thesis research. However, in Yu's model, they assume that one inbound door serves a single outbound door and the handling time is treated as input data.

Compared to Yu and Egbelu 2006's model, Boysen, et al. 2007 chose a more aggregate view, as detailed handling times of products are in general hard to obtain. The exact handling times for inbound trucks depend on the exact packing of goods and the sequence in which they can be obtained. Hence, they merge individual handling times for products to service slots to which inbound and outbound trucks are assigned.

Based on the former two papers, Shakeri, et al. 2008 present a detailed scheduling model combining the truck scheduling and truck-to-door assignment together to solve the scheduling problem. This involved processing time as well as loading, unloading and moving time. In this way, it is easier to estimate the job completion time. However, the processing time was not treated as a variable, which is a key difference from the approach in this thesis.

However, in real operations, Just-in-time scheduling policy is a very common consideration. Li, et al. 2004 and Alvarez-Perez, et al. 2009 presented optimization models using Just-in-time scheduling policy, which means every truck has a deadline for arrival and departure, and a penalty will occur if the due date of each truck is missed. The objective of the model is to schedule ITs and OTs to minimize the total earliness and tardiness of incoming and outgoing cargo, assuming a deadline of departure (for both ITs and OTs) in the form of a point in time.

In conclusion, in all the models presented, the handling time was treated as a given constant which is not realistic. In real life cross-docking operations, the handling time of each inbound truck depends on the location of itself and also the location of their outbound trucks it serves. A model using handling time as a randomly reasonable constant or just normalized data input would be not quite reliable to evaluate the whole cross-docking operation cost and time efficiency. Also, previous research has not considered the congestion that could occur inside the cross docking facilities which might cause temporary storage of goods (waiting time). Thus, the focus of this research is to develop a handling time function which defines it as a variable and includes all the time periods that may occur (e.g. loading, unloading time) based on truck-to-door assignment on both sets(ITs-IDs and OTs-ODs).

2.9 Contributions to Current Literature

In this thesis, two mathematical models are presented to solve truck scheduling problem at a cross-docking facilities.

Unlike two previous papers Li, et al. 2004; Alvarez-Perez, et al. 2009 using Just-in-time scheduling, the first model introduces departure requests in the form of a time window instead of a point of time. Penalty will occur if that departure time window deadline is not met. Another difference between the previous papers Li, et al. 2004; Alvarez-Perez, et al. 2009 and the current research is that we assume that truck handling time is not only dependent on the amount of cargo (un)loaded but also on the truck-to-door assignment. In the model presented herein truck handling time is a function of the distance the forklifts will travel carrying cargo from the ITs to the OTs. Further details on this assumption will be provided in the next section (i.e. model assumptions and formulation). This is the first time in the published literature that such a truck scheduling policy under these assumptions has been presented, thus is the key contribution of this work. However, the mathematical formulation is still NP-Complete, even in the simple case of one ID and one OD. To tackle this issue and solve the resulting problem with reasonable computational effort a memetic algorithm is developed and the scheduling policy and resolution algorithm are evaluated through a number of numerical examples.

Considering the congestion situation and the temporary storage which could happen, a bi-objective model to assign incoming and outgoing trucks to the inbound and outbound doors with two objectives: a) minimization of the total service time for all the trucks served at the facility, and b) minimization of the total storage time of the commodities transferred from the incoming to the outgoing trucks is also proposed. This approach can be considered as a relaxed version of the zero-inventory policy (Boysen 2010) that avoids infeasibility issues of processing the required number of outbound trucks (i.e. need to serve simultaneously a larger number of OTs than the available ODs) or avoid increased waiting times of the inbound trucks (i.e. use truck as a storage area while waiting for the outbound truck to start service). In this model it is also assumed that truck handling times are not only dependent on the amount of cargo (un)loaded but also on the truck-to-door assignment. Thus, truck handling time here is considered as a function of the distance forklifts will travel carrying cargo from the ITs to the OTs (this assumption is further discussed in the section 3 and 5 of model assumptions and formulation). In this way, an estimation of all the time periods could occur inside the cross-docking facility so that a temporary storage time is obtained. To tackle this issue and solve the resulting problem with reasonable computational effort a multi-objective memetic algorithm is developed and presented.

A limitation of our models is that it does not include the scheduling of the forklifts serving the ITs and OTs. In reality, the handling time of a truck is affected both by the truck-to-door assignment (both the ITs and OTs) and by the number of forklifts assigned to (un)load and move the cargo within the facility. For example: the assignment of more forklifts to a truck will reduce its handling time and this can compensate for an assignment to an ID further away from the ODs, or from the storage area where the cargo will be unloaded. On the other hand, an increase in the number of forklifts will increase costs and it might increase congestion within the facility, slowing down the speed of the forklifts and thus impeding the (un)loading and storage operations. The simultaneous scheduling of trucks-to-doors and forklifts-to-trucks is left to future research. In this thesis we assume that a sufficient number of forklifts are available so that trucks do not have to wait while at the IDs and ODs.

In conclusion, the focus of this thesis is to solve truck scheduling problem to get an optimized truck-to-door assignment result. Two mathematical models are presented to solve the problem with the objectives of minimizing total service time, minimizing total cost from tardy and early departures and minimizing total storage time. And two heuristics are designed to solve these two models. In both models formulations, handling time inside the facility is treated as a variable function of the door assignment of both sets of trucks. This is the key contribution of this research, as prior research has only considered the handling time as a input constant. This formulation makes these models more realistic so that an assignment more close to real life operations can be obtained, and fills a gap that exists in current published literature on this topic.

3. TRUCK SCHEDULING AT A CROSS-DOCKING FACILITY: MINIMIZING COSTS AND MAXIMIZING THROUGHPUT

3.1 Model Assumptions

The scheduling of ITs and OTs to the IDs and ODs of a facility can be formulated as the flowshop machine scheduling problem (FMSP) (Chen, et al. 2009) where we consider a set n of independent and non-preemptive jobs (i.e. ITs and OTs) to be processed on two sets of m unrelated machines in series (i.e. IDs and ODs). Each job may be processed on any of the m machines, but the processing time depends on the machine that executes the job. In the setup of a cross dock facility the processing time of an IT consists of the unloading time at the door and the travel time of the unloading equipment from the ID to the staging area or to the OD. The processing time of an OT consists of the loading time at the door and the travel time of the loading equipment from the staging area or from the ID. Under ideal conditions OTs would be scheduled for service at ODs opposite from the IDs that ITs with cargo for them are served (Figure 3.1a). As this distance increases so does the handling time of the trucks (ITs, OTs or both), mainly due to the increase of the forklift travel time between the IDs and ODs (Figure 3.1b). These conditions do not change even if cargo is temporarily stored within the facility (i.e. two-touch complexity shown in Figure 3.1c,d). In the model presented herein the handling time of both the ITs and OTs is a variable function of the door assignment of both sets of trucks. In this thesis we also assume that each truck (inbound or outbound) makes a time window request for departing from the facility. If the truck departs before or after this time window, the facility operator is penalized (similar to (Li, et al. 2004; Alvarez-Perez, et al. 2009)).

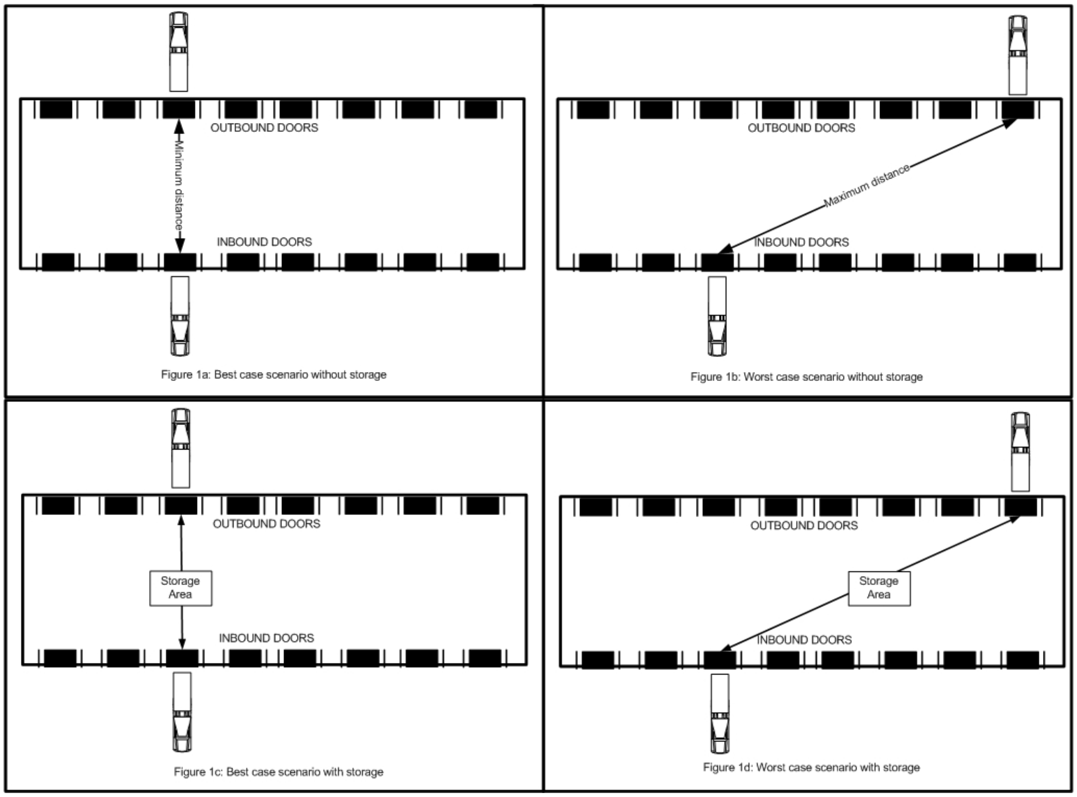

Figure 3.1 Best and Worst Case Truck-to-Door Assignments

3.2 Model Formulation

In order to formulate the problem of truck scheduling, under these assumptions, so as to minimize total service time, and total cost from tardy and early departures, we define the following:

Sets

I_1, I_2: set of inbound and outbound doors

J_1, J_2: set of inbound and outbound trucks

Decision Variables

$x_{ij}, i \in I_1, I_2, j \in J_1, J_2$ =1 if truck j is served at door i and zero otherwise

$y_{ab}, a, b \in J_1, J_2$	=1 if truck *b* is served at the same door as truck *a* as its immediate successor and zero otherwise
$f_j, \in J_1, J_2$	=1 if truck *j* is served as the first truck (at the door it is assigned) and zero otherwise
$l_j, j \in J_1, J_2$	=1 if truck *j* is served as the last truck (at the door it is assigned) and zero otherwise

Auxiliary Variables

$LD_j, j \in J_1, J_2$	minutes of late departure of truck *j*
$ED_j, j \in J_1, J_2$	minutes of early departure of truck *j*
$t_j, j \in J_1, J_2$	start time of service for truck *j* at its assigned door
$c_j\, j \in J_1, J_2$	handling time of truck *j*
$\Pi_j\, j \in J_2$	continuous positive variable number

Parameters

$LR_j, j \in J_1, J_2$	latest requested departure time of truck *j*
$ER_j, j \in J_1, J_2$	earliest requested departure time of truck *j*
$F_{ab}, a \in I_1, b \in I_2$	moving time of one unit forklift from door *a* to door *b*(in minutes)
$U_{ab}, a \in J_1, b \in J_2$	quantity of commodity carried by inbound truck *a* going to truck *b* (in forklift units)
$K_{ab}, a \in J_1, b \in J_2$	1 if incoming truck *a* carries cargo to be shipped out by outgoing truck *b* and zero otherwise
$A_j, j \in J_1, J_2$	arrival time of truck *j*

$S_i, i \in I_1, I_2$	time door i becomes available for the first time in the planning horizon[1]
$a_j, j \in J_1, J_2$	cost per minute of early departures
$b_j, j \in J_1, J_2$	cost per minute of tardy departures
tl	loading time for one unit of commodity
tu	unloading time for one unit of commodity
M	large positive number
N_1, N_2	normalizing factors (positive numbers)

The model formulation (from now on referred to as Model 1) minimizing the total cost from tardy and early departures and optimizing for the total throughput can be formulated as follows:

$$\min\left[\frac{\sum_{j\in J_1,J_2}(t_j - A_j) + \sum_{i\in I_1,I_2}\sum_{j\in J_1,J_2} c_j x_{ij}}{N_1} + \frac{\sum_{j\in J_1,J_2} b_j LD_j + \sum_{j\in J_1,J_2} a_j ED_j}{N_2}\right] \tag{1}$$

Subject To:

$$\sum_{i\in I_1,I_2} x_{ij} = 1, \forall j \in J_1, J_2 \tag{2}$$

$$f_b + \sum_{a\in J_1,J_2 \neq b} y_{ab} = 1, \forall b \in J_1, J_2 \tag{3}$$

$$l_a + \sum_{b\in J_1,J_2 \neq a} y_{ab} = 1, \forall a \in J_1, J_2 \tag{4}$$

[1] Some doors may not be available at time zero (i.e. start of planning horizon) as they may be still serving trucks from the previous planning horizon

$$f_a + f_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a, b \in J_1, a \neq b \tag{5}$$

$$l_a + l_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a, b \in J_1, a \neq b \tag{6}$$

$$y_{ab} - 1 \leq x_{ia} - x_{ib} \leq 1 - y_{ab}, \forall i \in I_1, a, b \in J_1, a \neq b \tag{7}$$

$$f_a + f_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a, b \in J_2, a \neq b \tag{8}$$

$$l_a + l_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a, b \in J_2, a \neq b \tag{9}$$

$$y_{ab} - 1 \leq x_{ia} - x_{ib} \leq 1 - y_{ab}, \forall i \in I_2, a, b \in J_2, a \neq b \tag{10}$$

$$t_j \geq A_j \forall j \in J_{1,} J_2 \tag{11}$$

$$t_j \geq S_i f_j \forall j \in J_{1,} J_2, i \in I_1, I_2 \tag{12}$$

$$t_b \geq t_a + \sum_{i \in I_1} c_a x_{ia} - M(1 - y_{ba}), \forall a, b \in J_1, a \neq b \tag{13}$$

$$t_b \geq t_a + \sum_{i \in I_2} c_a x_{ia} - M(1 - y_{ba}), \forall a, b \in J_2, a \neq b \tag{14}$$

$$ED_j \geq ER_j - t_j - \sum_{i \in I_1} c_j x_{ij}, \forall j \in J_1 \tag{15}$$

$$ED_j \geq ER_j - t_j - \sum_{i \in I_2} c_j x_{ij}, \forall j \in J_2 \tag{16}$$

$$LD_j \geq t_j + \sum_{i \in I_1} c_j x_{ij} - LR_j, j \in J_1 \tag{17}$$

$$LD_j \geq t_j + \sum_{i \in I_2} c_j x_{ij} - LR_j, j \in J_2 \tag{18}$$

$$c_j \geq \sum_{j'} U_{jj'} K_{jj'} \left(\sum_a \sum_b (F_{ab} x_{aj} + tu) \right) - M(1 - y_{bj'}), \forall a \in I_1, b \in I_2, j \in J_1, j' \in J_2 \tag{19}$$

$$\Pi_j \geq (t_i + c_i - t_j) K_{ij}, \forall i \in J_1, j \in J_2 \tag{20}$$

$$c_j \geq \sum_{j'} U_{jj'} K_{jj'} \left(\sum_a \sum_b (F_{ab} x_{aj'} + tl) \right) - M(1 - y_{bj}) - \Pi_j, \forall a \in I_1, b \in I_2, j \in J_2, j' \in J_1 \tag{21}$$

$$y_{ab} = 0, \forall a \in J_1, b \in J_2 \tag{22}$$

$$y_{ab} = 0, \forall b \in J_1, a \in J_2 \tag{23}$$

$$x_{ab} = 0, \forall a \in I_2, b \in J_1 \tag{24}$$

$$x_{ab} = 0, \forall a \in I_1, b \in J_2 \tag{25}$$

$$x_{ij} \in \{0,1\}, \forall i \in I_1, I_2 j \in J_1, J_2 \tag{26}$$

$$y_{ab} \in \{0,1\}, \forall a, b \in J_1, J_2, a \neq b \tag{27}$$

$$f_j, l_j \in \{0,1\}, \forall j \in J_1, J_2 \tag{28}$$

$$t_j, LD_j, ED_j, \Pi_j \in R^+, \forall \in J_1, J_2 \tag{29}$$

The objective function minimizes the total service time for all the trucks (first component) and the total cost from early and late departures for all the trucks (second component). N_1 and N_2 are normalizing factors obtained by solving two single objective optimization problems (see formulations SO_1 and SO_2) given the same feasible space as the original problem. Constraint set (2) ensures that each IT and OT are only served once. Constraint sets (3) and (4) ensure that each IT and OT will either be served first or be preceded by another truck. In a similar manner constraint sets (5) through (7) ensure that each IT and OT will either be served last or will be served before another truck. Constraint sets (8) though (10) ensure that only one IT can be served first and last at each door. Constraint set (11) forces a truck to start service after its arrival and after the door becomes available for the first time in the planning horizon (if the truck is served as the first truck). Constraint sets (13) and (14) estimate the start time of the inbound and outbound trucks. Constraints set (15) through (18) estimate the minutes of late and early departure. Constraint sets (19) through (21) estimate the handling time of the inbound and outbound trucks. The handling time of the IT is equal to the unloading time and the time it takes to move the products from the ID to the ODs where the OTs receiving cargo from this IT are assigned. The handling time of the OT is equal to the time it requires to transfer and load all the commodities from the IDs reduced by the time that the ITs are served before the OT starts service. Constraint sets (22) through (27) ensure that an IT will never be served at an OD and vice versa. Finally, equations (28) through (29) define the range of the decision and auxiliary variables.

The objective function minimizes the total service time for all the trucks (first component) and the total cost from early and late departures for all the trucks (second component). To solve this as a single objective model, we need to introduce normalizing factors:

$$\begin{aligned}
&SO_1 : N_1 = f_1(x) \\
&where: \\
&x = \arg\min f_1(x) \\
&s.t. \\
&(2)-(13),(18)-(28) \\
&f_1(x) = \sum_{j \in J_1, J_2} (t_j - A_j) + \sum_{i \in I_1, I_2} \sum_{j \in J_1, J_2} c_j x_{ij}
\end{aligned}$$

$$\begin{aligned}
&SO_2 : N_2 = f_2(x) \\
&where: \\
&x = \arg\min f_2(x) \\
&s.t. \\
&(2)-(28) \\
&f_2(x) = \sum_{j \in J_1, J_2} b_j LD_j + \sum_{j \in J_1, J_2} a_j ED_j
\end{aligned}$$

N_1 and N_2 are normalizing factors obtained by solving two single objective optimization problems (see formulations SO_1 and SO_2) given the same feasible space as the original problem.

4. SOLUTION ALGORITHM FOR MODEL 1

The problem formulation presented in section 3 is *NP-Complete* as it can be reduced to the Multi-Traveling Salesman Problem (MSTP). It is thus highly unlikely that an exact resolution algorithm exists that can solve real life instances of the problem in tractable computational times. In order to overcome this obstacle, a multi-population Memetic Algorithm (MA) was created. MAs are local optimization stochastic heuristics that combine the search attributes of Evolutionary Algorithms (EAs) with local search to improve the individual solutions. The common idea behind MAs and EAs is closely related to neighborhood search heuristics with the addition that at each step of the search multiple regions of the feasible space are visited. In general both MAs and EAs create randomly (or based on a rule) a set of candidate solutions that are recombined over a series of iterations. At each iteration, after the recombination step, and given a fitness function (that can be different from the objective function) candidate solutions with better values for the fitness function are selected to move on to the next iteration. This procedure is iterated until a candidate solution meets certain criteria (usually non-improvement of the fitness function value over a period of iterations) or an a-priori set computational limit is reached (usually CPU time or number of iterations). The main difference between MAs and EAs is that at each or some iteration(s) some or all of the candidate solutions are improved via the use of a local search heuristic using the same objective function as in the evolutionary counterpart. For more information we refer the interested reader to Moscato 1999; Hart, et al. 2008. In the remainder of this section, a detailed description of the proposed MA constructed to solve the problem at hand is provided. The MA presented herein was based on a Genetic Algorithms (GAs) heuristic (specific type of EA) proposed by Golias, et al. 2009.

Before continuing with the description of the MA, two definitions by Nguyen, et al. 2007, used within this thesis, are presented for consistency purposes:

Definition 1: Individual learning frequency, f_{il}, is defined as the proportion of an EA population that undergoes individual learning. For instance, if po is the EA or MA population size, the number of individuals in the population that undergoes individual improvement is then $f_{il} \times po$.

Definition 2: Individual learning intensity, t_{il}, is defined as the amount of computational budget allocated to an iteration of individual learning.

4.1 Chromosomal Representation

In scheduling problems, similar to the one presented here, integer chromosomal representation is more adequate (Eiben A.E. and Smith J.E. 2003; Boilé, et al.2009) and is thus adopted. An illustration of the chromosome structure, used in this thesis, is given in Figure 4.1 for a small instance of the problem with 6 inbound and 6 outbound trucks, and 2 inbound and 2 outbound doors. As seen in Figure 4.1, the chromosome consists of two sub-chromosomes: one for the ITs and one for the OTs. In this example both sub-chromosomes have two rows of 6 cells (equal to the total number of ITs or OTs). The cells in the upper row denote the door assignment while the lower rows represent the truck and its order of service. For example, IT=2 will be served first at the first door, IT=4 will be served second at the first door etc. The initial population for our experiments was created based on the First Come First Served (FCFS) rule at the door with the Smallest Queue (FCFSSM).

	Chromosome for Inbound Trucks					
Door	1	1	1	2	2	2
Inbound Truck	2	4	1	5	6	3

	Chromosome for Outbound Trucks					
Door	1	1	1	2	2	2
Outbound Truck	6	2	3	4	5	1

Figure 4.1 Illustration of Chromosome Representation

4.2 Recombination

Two of the most common types of recombination techniques usually applied in multi-population heuristic scheduling algorithms are the insert and swap mutation, illustrated in Figure 4.2 for the same small examples used in Figure 4.1. Both these types of mutation have been proven successful as they resemble variable small neighborhood search heuristics. Crossover operations are not usually applied in these types of scheduling problems with this chromosomal representation as they create a large number of infeasible solutions that require additional computational time to become feasible (Boilé, et al. 2009)

Due to the nature of this problem (i.e. the truck handling time of both the inbound and outbound trucks is a function of the assignment of the trucks to the inbound and outbound doors) these common recombination operations might perform poorly as they do not account for the relationship of the truck handling time, the door assignment and the start time of service. For this reason instead of the mutation operations we perform a local search on each chromosome at each iteration in order to combine both the inbound and outbound chromosomes. The local search consists of two optimization problems (with the same objective function and constraints as the original problem presented in section 3) solved in series.

As was discussed in subsection 4.1, each chromosome consists of two separate sub-chromosomes: one for the ITs-to-IDs assignment and one for the OTs-to-ODs. During the local search and for the first optimization problem we optimize for the schedule of the ITs given the schedule of the OTs, at the current iteration, as input while for the second optimization problem we optimize the schedule of the OTs given the schedule of the ITs, at the current iteration. We set the learning frequency and learning intensity equal to: $f_{il} = 1$ and, $t_{il} = 500$ iterations. Although both values of these parameters are high, and will increase the total computational burden, they do improve the rate of convergence of the MA (as will be shown in the next section through the computational examples). As both of these optimization problems are NP-Complete the GAs based heuristic presented by (Golias, et al. 2009) for the unrelated machine scheduling problem is used as the algorithm for the local search. The GA uses the same representation, fitness function and selection operators as described herein, and insert and swap mutation for recombination.

	SWAP MUTATION						
Before	Door	1	1	1	2	2	2
	Inbound Truck	2	4	1	5	6	3
After	Door	1	1	1	2	2	2
	Inbound Truck	6	4	1	5	2	3

	INSERT MUTATION						
Before	Door	1	1	1	2	2	2
	Inbound Truck	2	4	1	5	6	3
After	Door	1	1	1	1	2	2
	Inbound Truck	2	5	4	1	6	3

Figure 4.2 Schematic Illustrations of the Typical Mutation Operations

4.3 Fitness Function and Selection

Since the problem is a minimization problem, the smaller the value of the objective function, the higher the fitness value. The fitness function proposed by (Goldberg 1989) was used. This is given by: $z_t^i(x) = \max_i(f_t^i(x)) - f_t^i(x)$, where $f_t^i(x)$ is the objective function value and $z_t^i(x)$ is the fitness function value of chromosome *i* at iteration *t* for each chromosome. To avoid trapping at local optimal locations a number of high and medium fitness solutions are selected probabilistically at each generation, using the roulette wheel selection algorithm (Goldberg 1989), to form the population of the next generation. The proposed MA is also shown in Figure 4.3 where the left part of the flowchart shows the MA and the right side the GA applied as the local search. The algorithm was assumed to have converged if more than 15,000 iterations were performed or the objective function did not improve for 500 consecutive iterations.

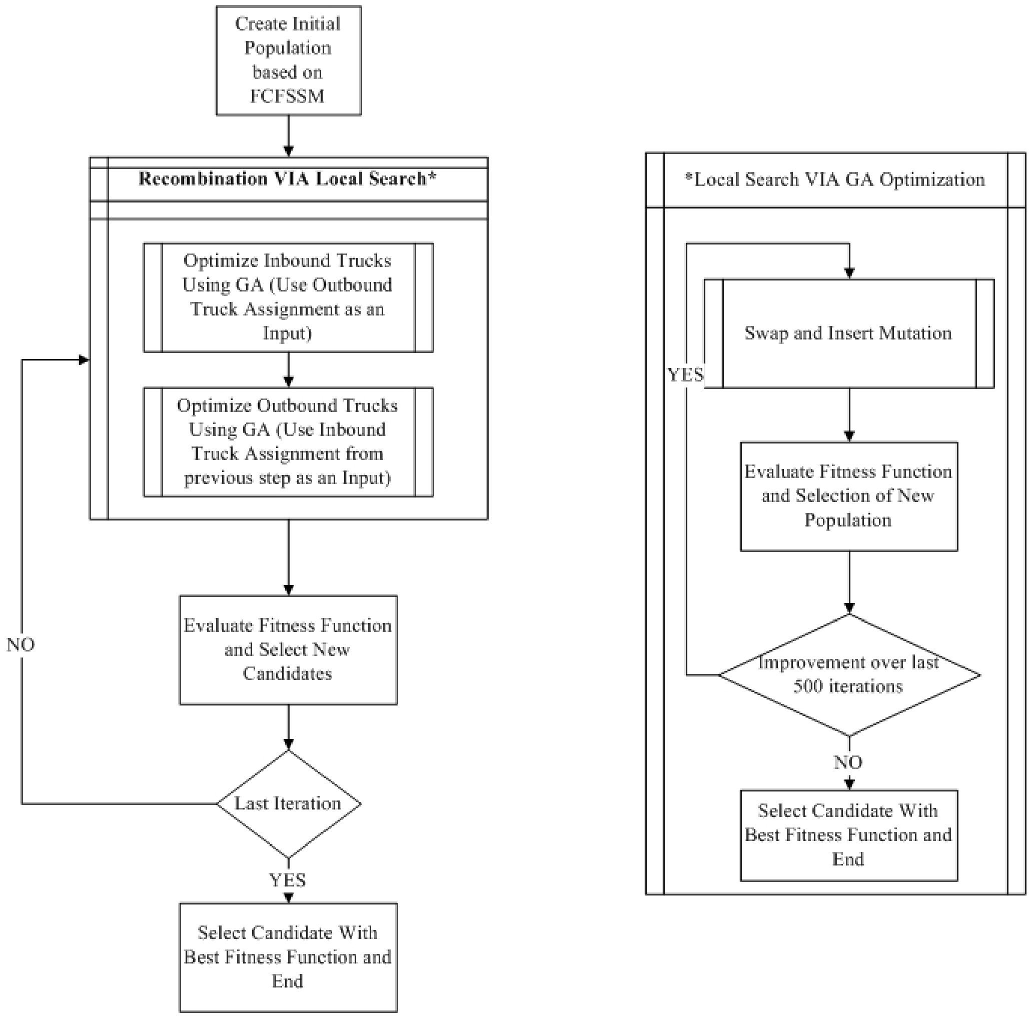

Figure 4.3 MA Flowchart

4.4 Computational Examples

10 datasets with three different truck inter-arrival times exponentially distributed (truncated to three times the mean) with a mean of five, ten, and fifteen minutes were created. We consider a facility with 10 IDs and 10 ODs and an 8 hour planning period. In total, 30 datasets were created. The handling time of a truck depends on the total cargo to be (un)loaded and the door assignment of both the ITs and OTs. The amount of inbound cargo destined to the OTs was created randomly (e.g. an IT arriving at the beginning of

the planning horizon can carry any amount of cargo for an OT arriving any time within the planning horizon[1]). The only restriction applied was that one IT could only carry cargo for a maximum of five out of the total number of OTs[2]. The travel time from the IDs to the ODs was estimated based on the geometry of the facility and a constant speed for the forklifts. A cross dock facility of an *I* shape, with the IDs and ODs on opposite sides of the facility was assumed. Each door (loading and unloading bay) was assumed to have a width of 15ft and the distance between two doors was assumed equal to 8 ft[3]. The width of the facility was assumed to be equal to 200ft (Figure 4.4). For more details on the dimensions of a cross dock facility we refer to (Drury and Falconer, 2003). It should be noted that the proposed model can be applied to any shape of a cross dock facility (e.g. *H*, *X* etc) as the shape of the facility only affects the input parameters and does not increase the complexity of the problem or the proposed resolution algorithm. The same statement is not true if congestion effects within the facility (as discussed in the introductory section) are to be included in the model. Modifying both the model and resolution algorithm to meet such assumptions is left as future research.

---

[1] From now on this assumption will be referred to as: ITs-to-OTs arrival time association

[2] From now on this assumption will be referred to as: ITs-to-OTs cargo association

[3] For fast turnaround a minimum of 13ft m bays are recommended with a 15 ft spacing preferable (*24*)

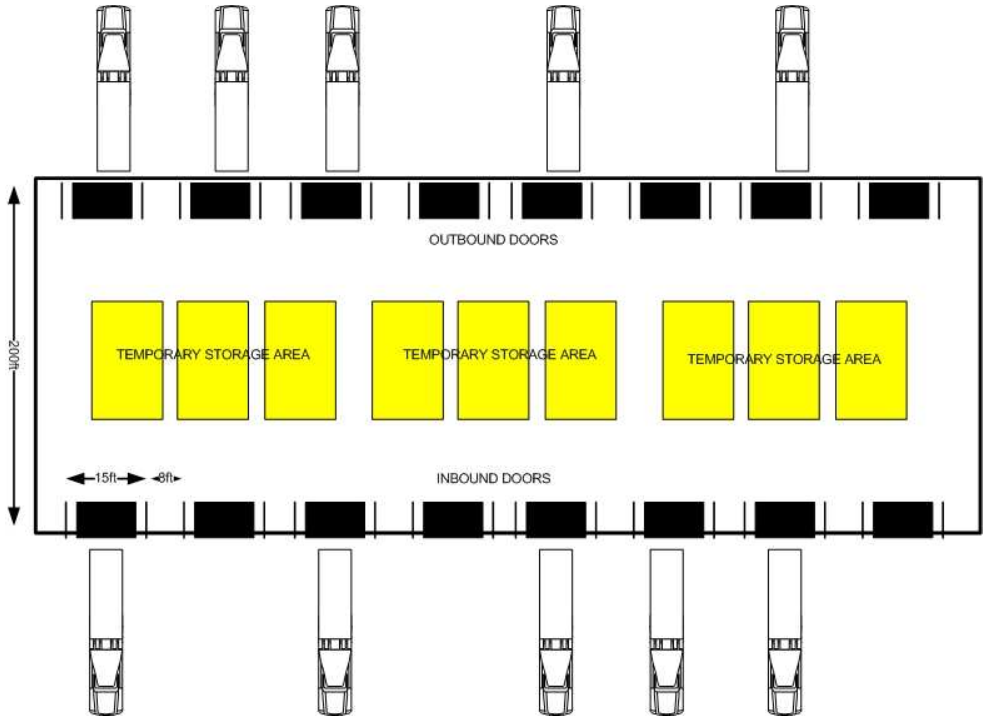

Figure 4.4 Facility Layout

Using the dataset described, four types of experiments were performed. The first was geared towards evaluating the influence of the learning frequency and intensity values to the objective function value and CPU time of the proposed MA. The second was geared towards evaluating the performance of the proposed resolution algorithm. The third type of experiment was geared towards the comparison of the proposed scheduling policy to the policy that schedules trucks sequentially (i.e. first ITs assuming an average handling time and then the OTs given the ITs assignment). The last type of experiment was geared towards: a) the performance of the proposed resolution algorithm, and b) the improvement of the proposed policy (over the policy that schedules trucks sequentially) to the assumptions of the ITs-to-OTs arrival time and cargo association. Results from these computational examples are presented in the following subsections.

4.5 MA Parameter Sensitivity

As was discussed in subsection 4.2 the parameters for the learning frequency and intensity were set to $f_{il} = 1$ and, $t_{il} = 500$. The influence of these parameters to the time required for the MA to converge (from now on referred to as convergence CPU time) and the objective function value was evaluated. Twenty-five different combinations of the learning and intensity frequency were created, with the former parameter values ranging from 0.1 to 1 (with an increment of 0.2) and the latter from 100 to 500 generations (with an increment of 100). For each one of these twenty-five combinations the 30 problem instances were solved and the change in the objective function and CPU time was recorded. The number of chromosomes selected to undergo the local search at each iteration where selected randomly using the roulette wheel selection with a selection probability for each chromosome equal to: $P(po_i \mid po) = {f(po_i)}\Big/{\max_i(f(po_i))}$, where P is the conditional probability that chromosome po_i will be selected for the local search at each iteration given the total population po, and $f(po_i)$ is the value of the objective function of chromosome po_i at the current iteration. As expected the objective function value decreased and the CPU time increased with the increase of the values of these parameters. The maximum difference in the CPU time between test instances with the smallest values (i.e. $f_{il} = 0.1$ and, $t_{il} = 100$) and the test instances with the largest values for the parameters (i.e. $f_{il} = 1$ and, $t_{il} = 500$) was less than 10 minutes, while the minimum difference in the objective function value was 10%. For this type of problem and for practical applications, 10 minutes is considered as a small computational expense.

Thus, the maximum values for both parameters can be used, so that the best objective function value is obtained.

4.6 MA Efficiency

In this subsection, a set of computational examples to evaluate the performance of the proposed MA is presented. As the problem presented herein is *NP-Complete*, optimal solutions could not be obtained even for small instances of the problem. Thus, to evaluate the performance of the heuristic, the convergence CPU time and the objective function values were compared to those obtained from the proposed MA and a variation of a MA proposed by Yeh 2002 (from now on referred to as MA2) for the flowshop scheduling problem. Heuristic MA2, as with any heuristic not created for the problem at hand, was applied with the following modifications:

a) initial solutions were obtained using the FCFSSM (also used by the MA),
b) crossover probabilities were decreased and set equal to zero (to avoid infeasibility issues),
c) mutation probabilities were increased and set equal to 1, and
d) 10 machines were used (instead of 2).

The 30 problem instances (presented in the beginning of this section) were solved using both metaheuristics (i.e. MA and MA2). From now on, the schedules obtained by the MA and MA2 will be referred to as S_1 and S_2 respectively. Table 4.1 summarizes the differences in the objective function values between the two schedules (S_1 and S_2) in %, while Table 4.2 shows the CPU time required for each one to fully converge. From the results in Tables 4.1 and 4.2, it is evident that MA was able to provide an improved objective function value up to 18% with reduced computational burden up to 76%. It should be noted that both heuristics CPU time was within acceptable limits for practical applications.

Table 4.1 Improvement of Objective Function Value (MA VS MA2)

	TRUCK INTER-ARRIVAL		
	5 min	10 min	15 min
Dataset	S_1 VS S_2	S_1 VS S_2	S_1 VS S_2
1	5%	17%	17%
2	6%	9%	15%
3	15%	9%	4%
4	11%	5%	9%
5	7%	8%	12%
6	5%	12%	7%
7	8%	9%	18%
8	8%	9%	12%
9	4%	9%	12%
10	6%	9%	14%

S_1: Schedule obtained from MA (until convergence)

S_2: Schedule obtained from MA2 (until convergence)

Table 4.2 Improvement of CPU Time Required for Full Convergence (MA VS MA2)

	TRUCK INTER-ARRIVAL		
	5 min	10 min	15 min
Dataset	S_1 VS S_2	S_1 VS S_2	S_1 VS S_2
1	29%	45%	72%
2	45%	39%	70%
3	45%	58%	43%
4	14%	60%	41%
5	43%	48%	76%
6	26%	53%	65%
7	20%	45%	70%
8	8%	33%	69%
9	35%	36%	65%
10	23%	70%	52%

S_1: Schedule obtained from MA (until convergence)

S_2: Schedule obtained from MA2 (until convergence)

4.7 Scheduling Policy Evaluation

The proposed truck-to-door scheduling policy (i.e. combined schedule of ITs and OTs) was compared to the truck-to-door scheduling policy where the ITs and OTs are scheduled sequentially (i.e. first schedule the ITs and then given the IT-to-ID assignment schedule the OTs to the ODs). The problem formulations of scheduling ITs and OTs sequentially are shown in Appendix A and B. In order to solve the sequential scheduling problem the handling time of each IT is required as an input. As previously discussed, this time depends on the assignment of both ITs-to-IDs and OTs-to-ODs which is not know in advance. To address this issue and solve the sequential scheduling problem, an average handling time for the ITs was used. This time was equal to the mean unloading time between the OD furthest and closest to each ID for each IT. For example (as shown in Figure 4.5) assume truck $a \in I_1$ is scheduled for service at door 3. The closest OD is door 3 and the furthest away OD is door 10. The time to unload all the cargo at ID=3 is equal to: $C_{a3} = \sum_b U_{ab} F_{33}, b \in J_2$ while the time to unload all the cargo to door 3 is equal to: $C_{a3} = \sum_b U_{ab} F_{3,10}, b \in J_2$. Here an average handling time of: $C_{a3} = \sum_b U_{ab} (\frac{F_{3,10} - F_{33}}{2}), a \in I_1, b \in I_2$ is assumed.

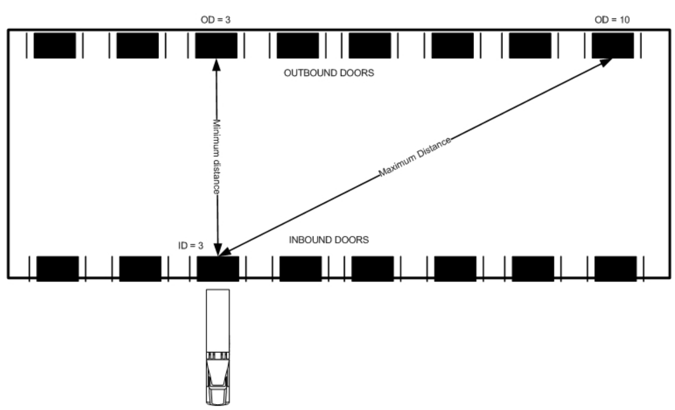

Figure 4.5 Example of Handling Time Estimation

Even with the assumption of known handling times for the ITs (based on the previous discussion), solving the sequential problem requires solving two *NP-Complete* problems (i.e. the truck-to-door assignment for the ITs and the OTs separately). To avoid bias in the results the proposed MA was applied as a solution algorithm, where at the recombination step the optimization of the OTs was omitted when the MA was used for scheduling the ITs and vice versa. Results from the computational examples are shown in Table 4.3. The first column shows the dataset number. Columns two through four show the differences (in %) of the objective between the schedules obtained using the two scheduling approaches (i.e. combined scheduling-CS and sequential scheduling-SS) for the datasets with truck inter-arrival times of five minutes. Column two shows the improvement in the total service time and delayed/early departures for the ITs, columns three and four show the same results but for the OTs, and for both the ITs and OTs. The remaining columns in Table 4.3 show the same results for the datasets with truck inter-arrival times of ten and fifteen minutes. For example, for the first dataset and 5 minutes of truck inter-arrival time, the schedule obtained using the proposed policy has an improvement[4] of 23%, 8%, and 15% for the ITs, OTs, and for both the ITs and OTs. Looking at the results for the remaining datasets and inter-arrival times, it is observed that the proposed scheduling policy always provides an improved schedule when compared to the schedule obtained using the sequential scheduling approach.

[4] Improvement in the objective function value

Table 4.3 Differences in The Objective Function Between The CS and The SS Approach (no restriction)

	5min			10min			15min		
Dataset	**Inbound**	**Outbound**	**Total**	**Inbound**	**Outbound**	**Total**	**Inbound**	**Outbound**	**Total**
1	23%	8%	15%	47%	13%	25%	45%	9%	22%
2	24%	11%	16%	46%	3%	19%	50%	17%	26%
3	32%	12%	21%	29%	10%	16%	19%	5%	9%
4	30%	12%	19%	21%	9%	12%	20%	8%	12%
5	29%	3%	14%	32%	7%	16%	35%	18%	24%
6	10%	2%	5%	40%	7%	20%	36%	14%	22%
7	22%	8%	14%	42%	15%	24%	33%	17%	23%
8	28%	11%	17%	31%	7%	16%	39%	11%	21%
9	17%	5%	10%	37%	17%	24%	28%	13%	17%
10	18%	2%	8%	44%	12%	23%	35%	12%	19%

4.8 ITs and OTs arrival time association

In the initial data assumptions, ITs were allowed to carry cargo for any OT irrelevant of its arrival time. In this subsection, the effect of this relationship (i.e. the arrival time of the OTs and the arrival time of ITs carrying their cargo) on the efficiency of the MA (in terms of the convergence CPU time) and the improvement of the objective function value of schedule S_1 over schedule S_2 is evaluated. Using the assumptions for the data presented in the beginning of this section, 10 additional datasets for each inter-arrival time (i.e. 5, 10, and 15 minutes) were created. Unlike the datasets used in the experiments in the subsection 4.7 (where ITs could carry cargo for any OT irrelevant of its arrival time), in these datasets ITs were restricted in carrying cargo for OTs arriving at a maximum of four and six hours later (i.e. an IT arriving at time t could only carry cargo for OTs arriving at time less than $t+4$ hours and $t+6$ hours respectively). Results from these experiments are shown in Tables 4.4 and 4.5 (4 and 6 hours arrival time restriction respectively) for the combined and sequential scheduling models (similar to results presented in Table 4.3). From these tables, it is evident that the benefits remain significant. The convergence CPU time improved slightly with the new datasets, with the maximum reduction reaching 10% when compared to the convergence CPU time of the examples in subsection 4.7.

Table 4.4 Differences in The Objective Function Between The CS and the SS Approach (4 hour restriction)

	5min			10min			15min		
Dataset	**Inbound**	**Outbound**	**Total**	**Inbound**	**Outbound**	**Total**	**Inbound**	**Outbound**	**Total**
1	15%	35%	13%	35%	18%	23%	38%	17%	24%
2	24%	34%	17%	34%	9%	17%	40%	22%	28%
3	17%	39%	9%	39%	15%	22%	35%	8%	17%
4	25%	33%	14%	33%	17%	22%	38%	14%	20%
5	24%	35%	17%	35%	20%	26%	2%	4%	4%
6	18%	49%	13%	49%	21%	31%	48%	21%	29%
7	20%	30%	13%	30%	9%	17%	38%	24%	28%
8	24%	33%	12%	33%	14%	22%	37%	20%	25%
9	26%	46%	13%	46%	21%	30%	42%	20%	29%
10	30%	33%	19%	33%	18%	23%	27%	10%	15%

Table 4.5 Differences in The Objective Function Between The CS and The SS Approach (6 hour restriction)

	5min			10min			15min		
Dataset	**Inbound**	**Outbound**	**Total**	**Inbound**	**Outbound**	**Total**	**Inbound**	**Outbound**	**Total**
1	22%	37%	38%	12%	29%	17%	37%	37%	25%
2	36%	20%	38%	39%	37%	35%	26%	34%	23%
3	33%	39%	22%	25%	25%	20%	30%	32%	22%
4	10%	14%	34%	34%	34%	38%	36%	24%	18%
5	20%	37%	26%	23%	27%	29%	20%	30%	14%
6	37%	19%	13%	20%	24%	27%	22%	24%	38%
7	29%	39%	15%	27%	21%	22%	25%	25%	20%
8	25%	20%	37%	24%	36%	29%	31%	13%	23%
9	20%	32%	16%	11%	31%	28%	16%	19%	25%
10	38%	32%	25%	26%	32%	23%	38%	16%	11%

4.9 ITs and OTs cargo association

In the initial data assumptions, ITs were allowed to carry cargo for a maximum of five OTs. In this subsection, the effect of this relationship (i.e. the maximum number of OTs that an IT can carry cargo for) on the efficiency of the MA (in terms of the convergence CPU time) and the improvement of the objective function value of schedule S_1 over schedule S_2 is evaluated. In the datasets used in the experiments, ITs were allowed to carry cargo for a maximum of ten and fifteen OTs. Results from these experiments are shown in Tables 4.6 and 4.7 for the combined and sequential scheduling models (similar to results presented in Table 4.3, 4.4 and 4.5). From these tables, it is observed that the benefits remain significant. Unlike results in subsection 4.8, the convergence CPU time for these datasets almost doubled (on average) compared to the convergence CPU time of the examples in subsection 4.8, reaching a maximum of 20 minutes (for the examples with the five minutes truck inter-arrival time). This increase in the computational burden is acceptable as it is less than the average handling time of an average truck (Boysen 2010).

Table 4.6 Differences in The Objective Function Between The CS and The SS Approach (One IT carrying cargo for a maximum of 10 OTs)

	5min			**10min**			**15min**		
Datase t	**Inboun d**	**Outboun d**	**Tota l**	**Inboun d**	**Outboun d**	**Tota l**	**Inboun d**	**Outboun d**	**Tota l**
1	24%	11%	17%	37%	22%	26%	22%	9%	13%
2	18%	6%	11%	35%	17%	24%	37%	14%	21%
3	15%	3%	9%	46%	27%	36%	32%	21%	24%
4	26%	10%	17%	27%	7%	14%	42%	27%	32%
5	30%	16%	21%	23%	14%	17%	53%	28%	38%
6	28%	10%	16%	35%	21%	26%	41%	22%	29%
7	27%	10%	16%	30%	12%	18%	41%	18%	26%
8	30%	10%	18%	38%	19%	26%	31%	14%	19%
9	22%	6%	12%	28%	15%	20%	38%	18%	26%
10	19%	12%	16%	45%	22%	30%	12%	8%	9%

Table 4.7 Differences in the objective function between the CS and the SS approach (One IT carrying cargo for a maximum of 15 OTs)

	5min			10min			15min		
Datase t	**Inboun d**	**Outboun d**	**Tota l**	**Inboun d**	**Outboun d**	**Tota l**	**Inboun d**	**Outboun d**	**Tota l**
1	25%	9%	16%	29%	21%	24%	30%	18%	22%
2	23%	11%	8%	32%	14%	20%	45%	21%	28%
3	23%	7%	13%	28%	13%	18%	43%	13%	25%
4	19%	9%	8%	39%	12%	19%	21%	13%	15%
5	22%	12%	16%	32%	11%	16%	36%	15%	21%
6	27%	12%	18%	33%	13%	19%	25%	14%	17%
7	16%	2%	8%	43%	18%	27%	55%	23%	35%
8	18%	2%	9%	41%	12%	22%	20%	8%	13%
9	23%	1%	11%	37%	13%	21%	43%	17%	24%
10	24%	11%	17%	38%	20%	26%	31%	11%	16%

5. TRUCK SCHEDULING AT A CROSS-DOCKING FACILITY: MINIMIZING TOTAL SERVICE TIME AND TOTAL STORAGE TIME

5.1 Model Assumptions

The scheduling of ITs and OTs to the IDs and ODs of a facility can be formulated as the flowshop machine scheduling problem (FMSP) (Chen, et al. 2009) where a set n of independent and non-preemptive jobs (i.e. ITs and OTs) to be processed on two sets of m unrelated machines in series (i.e. IDs and ODs) is considered. Each job may be processed on any of the m machines, but the processing time depends on the machine that executes the job. In the setup of a cross-dock facility the processing time of an IT consists of the unloading time at the door and the travel time of the unloading equipment from the ID to the staging area or to the OD. The processing time of an OT consists of the loading time at the door and the travel time of the loading equipment from the staging area or from the ID. Under ideal conditions OTs would be scheduled for service at ODs opposite of the IDs that ITs with cargo for them are served (Figure 3.1-a). As this distance increases, so does the handling time of the trucks (ITs, OTs or both), mainly due to the increase in forklift travel time (Figure 3.1-b). These conditions do not change even if cargo is temporarily stored within the facility (i.e. two-touch complexity shown in Figure 3.1-c, 3.1-d). In the present model the handling time of both ITs and OTs is a function of the door assignment of both sets of trucks. Each IT is assigned a number of forklifts equal to the number of pallets that it carries (assuming that each forklift can carry one pallet at a time).

5.2 Model Formulation

To formulate the bi-objective problem of truck scheduling at the available doors under these assumptions, with the objective to minimize the total service time for all the trucks and minimize the total storage time for the inbound and outbound cargo, the followings are defined:

Sets

I_1, I_2:	set of inbound and outbound doors
J_1, J_2:	set of inbound and outbound trucks

Decision Variables

$x_{ij} \in \{0,1\} \forall i \in I_1, I_2, j \in J_1, J_2$	=1 if truck j (IT or OT) is served at door i and zero otherwise
$y_{ab} \in \{0,1\} \forall a, b \in J_1, J_2$	=1 if truck b (IT or OT) is served at the same door as truck (IT or OT) a as its immediate successor and zero otherwise
$f_j \in \{0,1\} \forall j \in J_1, J_2$	=1 if truck j (IT or OT) is served as the first truck (at the door it is assigned) and zero otherwise
$l_j \in \{0,1\} \forall, j \in J_1, J_2$	=1 if truck j (IT or OT) is served as the last truck (at the door it is assigned) and zero otherwise

Auxiliary Variables

$t_j \in R^+, \forall, j \in J_1, J_2$	start time of service for truck j (IT or OT) at its assigned door
$c_j \in R^+, \forall j \in J_1, J_2$	handling time of truck j (IT or OT)
$\Pi_j \in R^+, j \in J_2$	continuous positive variable

$T_{ab} \in R^{+}, \forall a \in J_2, b \in J_1$ — total stay time in the facility of the commodity transferred from IT *a* to OT *b*

Parameters

$F_{ab}, a \in I_1, b \in I_2$ — moving time of one unit forklift from door *a* to door *b*(in minutes)

$U_{ab}, a \in J_1, b \in J_2$ — quantity of commodity carried by IT *a* going to OT *b* (in forklift units)

$K_{ab}, a \in J_1, b \in J_2$ — 1 if IT *a* carries cargo to be shipped out by OT *b* and zero otherwise

$A_j, j \in J_1, J_2$ — arrival time of truck *j*

$S_i, i \in I_1, I_2$ — time door *i* becomes available for the first time in the planning horizon[i]

$a_j, j \in J_1, J_2$ — cost per minute of early departures

$b_j, j \in J_1, J_2$ — cost per minute of tardy departures

tl — loading time for one unit of commodity

tu — unloading time for one unit of commodity

M — large positive number

N_1, N_2 — normalizing factors (positive numbers)

The bi-objective model formulation (from now on referred to as Model 2) minimizing the total service time and total storage time can be formulated as follows:

$$\min\left[\sum_{j\in J_1,J_2}(t_j - A_j) + \sum_{i\in I_1,I_2}\sum_{j\in J_1,J_2} c_j x_{ij}\right]$$

(30.1)

$$\min\left[\sum_{a\in J_2}\sum_{b\in J_1} T_{ab}U_{ab}\right]$$

(1.2)

Subject To:

$$\sum_{i\in I_1,I_2} x_{ij} = 1, \forall j \in J_1, J_2 \tag{31}$$

$$f_b + \sum_{a\in J_1,J_2\neq b} y_{ab} = 1, \forall b \in J_1, J_2 \tag{32}$$

$$l_a + \sum_{b\in J_1,J_2\neq a} y_{ab} = 1, \forall a \in J_1, J_2 \tag{33}$$

$$f_a + f_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a,b \in J_1, a \neq b \tag{34}$$

$$l_a + l_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a,b \in J_1, a \neq b \tag{35}$$

$$y_{ab} - 1 \leq x_{ia} - x_{ib} \leq 1 - y_{ab}, \forall i \in I_1, a,b \in J_1, a \neq b \tag{36}$$

$$f_a + f_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a,b \in J_2, a \neq b \tag{37}$$

$$l_a + l_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a,b \in J_2, a \neq b \tag{38}$$

$$y_{ab} - 1 \leq x_{ia} - x_{ib} \leq 1 - y_{ab}, \forall i \in I_2, a,b \in J_2, a \neq b \tag{39}$$

$$t_j \geq A_j \forall j \in J_{1,} J_2$$

(40)

$$t_j \geq S_i f_j \forall j \in J_{1,} J_2, i \in I_1, I_2 \tag{41}$$

$$t_b \geq t_a + \sum_{i \in I_1} c_a x_{ia} - M(1 - y_{ba}), \forall a, b \in J_1, a \neq b \tag{42}$$

$$t_b \geq t_a + \sum_{i \in I_2} c_a x_{ia} - M(1 - y_{ba}), \forall a, b \in J_2, a \neq b \tag{43}$$

$$c_j \geq \sum_{j^{'}} U_{jj'} K_{jj'} \left(\sum_a \sum_b (F_{ab} x_{aj} + tu) \right) - M(1 - y_{bj'}),$$
$$\forall a \in I_1, b \in I_2, j \in J_1, j^{'} \in J_2 \tag{44}$$

$$\Pi_j \geq (t_i + c_i - t_j) K_{ij}, \forall i \in J_1, j \in J_2 \tag{45}$$

$$c_j \geq \sum_{j^{'}} U_{jj'} K_{jj'} \left(\sum_a \sum_b (F_{ab} x_{aj'} + tl) \right) - M(1 - y_{bj}) - \Pi_j,$$
$$\forall a \in I_1, b \in I_2, j \in J_2, j^{'} \in J_1 \tag{46}$$

$$T_{ab} \geq (t_a - t_b - c_b x_{ib}) K_{ba}, \forall a \in J_2, b \in J_1, i \in I_1 \tag{47}$$

$$y_{ab} = 0, \forall a \in J_1, b \in J_2$$

(1948)

$$y_{ab} = 0, \forall b \in J_1, a \in J_2$$

(49)

$$x_{ab} = 0, \forall a \in I_2, b \in J_1$$

(501)

$$x_{ab} = 0, \forall a \in I_1, b \in J_2$$

(512)

$$x_{ij} \in \{0,1\}, \forall i \in I_1, I_2\, j \in J_1, J_2$$

(523)

$$y_{ab} \in \{0,1\}, \forall a, b \in J_1, J_2, a \neq b$$

(534)

The first objective function minimizes the total service time for all the trucks. The second objective function minimizes the total stay time for all the cargo. In this thesis, it is assumed that the storage time begins after the IT has left the facility and before the OT starts service (e.g. if an IT is unloading and half way through the OT receiving the cargo starts service then all the product from the IT to the OT have a storage time of zero). Constraint set (2) ensures that each IT and OT are only served once. Constraint sets (3) and (4) ensure that each IT and OT will either be served first or be preceded by another truck. In a similar manner constraint sets (5) through (7) ensure that each IT and OT will either be served last or it will be served before another truck. Constraint sets (8) though (10) ensure that only one IT can be served first and last at each door. Constraint set (11) forces a truck to start service after its arrival and after the door becomes available for the first time in the planning horizon (if the truck is served as the first truck). Constraint sets (13) and (14) estimate the start time of the inbound and outbound trucks. Constraint sets (15) through (17) estimate the handling time of the inbound and outbound trucks. The handling time of the IT is equal to the unloading time and the time it takes to move the products from the ID to the ODs where the OTs receiving cargo from this IT are assigned. The handling time of the OT is equal to the time it requires to transfer and load all the commodities from the IDs reduced by the time that the ITs are served before the OT starts service. Constraint sets (18) estimates the stay time of the cargo inside the facility. Constraint sets (19) through (24) ensure that an IT will never be served at an OD and vice versa.

6. SOLUTION ALGORITHM FOR MODEL 2

Existing exact resolution algorithms for bi-objective scheduling problems rely on iterative-type of procedures. These procedures employ exact algorithms to solve single objective problem formulations of the original bi-objective formulations. These algorithms cannot be efficiently applied to our problem, as the single objective formulation of the Model 2, considering either or both objectivesii, is NP-hard. Thus, solving repetitively a large number of these single objective problems to optimality would require very large computational times. A compromising solution would be to use a heuristic as the solution approach of the single objective problem but this approach would not guarantee optimality, and thus defeat the purpose of using such an approach. To address this issue a Multi-Objective Memetic Algorithm (MOMA) is proposed that can handle any realistic size problem. Memetic Algorithm (MAs) are local optimization stochastic heuristics that combine the search attributes of Evolutionary Algorithms (EAs) with local search to improve the individual solutions. The common idea behind MAs and EAs is closely related to neighborhood search heuristics with the addition that, at each step of the search, multiple regions of the feasible space are visited. In general, both MAs and EAs create randomly (or based on a rule) a set of candidate solutions that are recombined over a series of iterations. At each iteration, after the recombination step, and given a fitness function (that can be different from the objective function), candidate solutions with better values for the fitness function are selected to move on to the next iteration. This procedure is iterated until a candidate solution meets certain criteria (usually non-improvement of the fitness function value over a period of iterations) or an a-priori set computational limit is reached (usually CPU time or number of iterations).

The main difference between MAs and EAs is that at each or some iteration(s), some or all of the candidate solutions are improved via the use of a local search heuristic using the same objective function as in the evolutionary counterpart. For more information we refer interested readers to (Moscato 1999) and (Hart, et al. 2005). In the remainder of this section we provide a detailed description of the proposed MA constructed to solve the problem at hand. The MA presented here is based on a Genetic Algorithms (GAs) heuristic (specific type of EA) proposed by (Golias, et al. 2009) and a single objective MA proposed by (Golias, et al. 2010).

Before continuing with the description of the MA, two definitions by (Nguyen, et al. 2003), used here, are presented for purposes of consistency:

Definition 1: Individual learning frequency, f_{il}, is defined as the proportion of an EA population that undergoes individual learning. For instance, if po is the EA or MA population size, the number of individuals in the population that undergoes individual improvement is then $f_{il} \times po$.

Definition 2: Individual learning intensity, t_{il}, is defined as the amount of computational budget allocated to an iteration of individual learning.

6.1 Chromosomal Representation

In scheduling problems, similar to the one presented here, integer chromosomal representation is more adequate (Eiben and Smith, 2003; Boile, et al. 2009; Wong and Leung, 2008) and is thus adopted. An illustration of the chromosome structure used here is given in Figure 6.1 for a small instance of the problem with 6 inbound and 6 outbound trucks, and 2 inbound and 2 outbound doors. As seen in Figure 6-1, the chromosome consists of two sub-chromosomes: one for the ITs and one for the OTs. In this example, both sub-chromosomes have two rows of 6 cells (equal to the total number of ITs or OTs). The cells in the upper row denote the door assignment while the lower rows represent the truck and its order of service. For example, IT=2 will be served first at the first door, IT=4 will be served second at the first door etc. The initial population for our experiments was created based on the First Come First Served rule at the door with the Smallest Queue.

	Chromosome for Inbound Trucks					
Door	1	1	1	2	2	2
Inbound Truck	2	4	1	5	6	3

	Chromosome for Outbound Trucks					
Door	1	1	1	2	2	2
Outbound Truck	6	2	3	4	5	1

Figure 6.1 Illustration of Chromosome Representation

6.2 Recombination

Two of the most common types of recombination techniques usually applied in multi-population heuristic scheduling algorithms are the insert and swap mutation, illustrated in Figure 6.2 for the same examples used in Figure 6.1. Both types of mutation have been proven successful as they resemble variable small neighborhood search heuristics. Crossover operations are not usually applied in these types of scheduling problems, with such chromosomal representation, as they create a large number of infeasible solutions that require additional computational time to become feasible (Boile, et al. 2009).

SWAP MUTATION							
Before	Door	1	1	1	2	2	2
	Inbound Truck	2	4	1	5	6	3
After	Door	1	1	1	2	2	2
	Inbound Truck	6	4	1	5	2	3

INSERT MUTATION							
Before	Door	1	1	1	2	2	2
	Inbound Truck	2	4	1	5	6	3
After	Door	1	1	1	1	2	2
	Inbound Truck	2	5	4	1	6	3

Figure 6.2 Illustration of the Typical Mutation Operations

Common recombination operations might perform poorly, as they do not account for the relationship between truck handling time, door assignment and the start time of service of the trucks (both ITs and OTs). For this reason at each iteration, instead of the mutation operations, we perform a local search on each combination of chromosomes, in order to combine both the inbound and outbound chromosomes. The local search consists of two optimization problems, with the same objective function and constraints as the original problem presented in the previous section, solved in series. These mutations are shown in Figure 6.3.

Procedure 1

$p=1,2,.....i$ population of inbound chromosomes
$q=1,2,,,,,,j$ population of outbound chromosomes
Step 0: Set i=j=0;
Step 1: Set i=i+1 and select inbound chromosome i
Step 2: If i = = $|p|$+1 end else go to step 2
Step 2: Set j=j+1
Step 3: Optimize inbound chromosome i for outbound chromosome j using local search
Step 4: If j = = $|q|$ go to step 1 else go to step 2

Procedure 2

$p=1,2,.....i$ population of outbound chromosomes
$q=1,2,,,,,,j$ population of inbound chromosomes
Step 0: Set i=j=0;
Step 1: Set i=i+1 and select outbound chromosome i
Step 2: If i = = $|p|$+1 end else go to step 2
Step 2: Set j=j+1
Step 3: Optimize outbound chromosome i for inbound chromosome j using local search
Step 4: If j = = $|q|$ go to step 1 else go to step 2

Figure 6.3 Mutation Procedures for Inbound and Outbound Chromosomes

As previously discussed, each chromosome consists of two separate sub-chromosomes: one for the IT-to-ID and one for the OT-to-OD assignments. During the local search, and for the first optimization problem, we optimize for the schedule of the ITs given the schedule of the OTs (at the current iteration for each outbound chromosome) as input, while for the second optimization problem we optimize the schedule of the OTs given the schedule of the ITs, at the current iteration. We set the learning frequency and learning intensity equal to: $f_{il} = 1$ and, $t_{il} = 500$ iterations. Although both values of these parameters are high, and will increase the total computational burden, they do improve the rate of convergence of the MA (as will be shown in the next section through the computational examples). As both of these optimization problems are NP-Hard, the GAs based heuristic presented by (Golias, et al. 2009) for the unrelated machine scheduling problem is used as the algorithm for the local search. The GA uses the same representation, fitness function as described here, and insert and swap mutation for recombination. The Roulette Wheel Selection (RWS) proposed by (Goldberg 1989) is applied.

Fitness Function and Selection: Since the problem is a minimization problem, the smaller the value of each objective function, the higher the fitness value. We use the fitness function proposed by (Goldberg 1989). This is given by: $z_t^i(x) = \max_i(f_t^i(x)) - f_t^i(x)$, where $f_t^i(x)$ is the objective function value and $z_t^i(x)$ is the fitness function value of chromosome i at iteration t for each chromosome. Once the fitness function has been estimated for both objective functions for all the chromosomes the solutions are ordered using the non-numerical ranking preferences method (NRPM) proposed by (Golias, et al. (in print)). At every generation the mutated population is copied into two sets. The first set is used to select parents for the next generation based on the optimal Pareto front. This selection technique retains variety in the selected solution. If the selected parents from this set are less than the initial population their number is increased by randomly copying from the newly selected parents. If the selected parents are more than the initial population their number is decreased to the initial population, using the RWS using their order number as the criterion of selection. The second set is used in an elitist way to obtain better minimum values for each objective function within the Pareto front; thus the best two chromosomes based on each fitness function value, are selected. Both sets are then combined into one. The selection procedure is shown in Figure 6.4. The proposed MOMA is shown in Figure 6.5 where the left part of the flowchart shows the MA and the right side the local search GA. The algorithm is assumed to have converged if more than 15,000 iterations are performed or the Pareto front does not improve for 500 consecutive iterations.

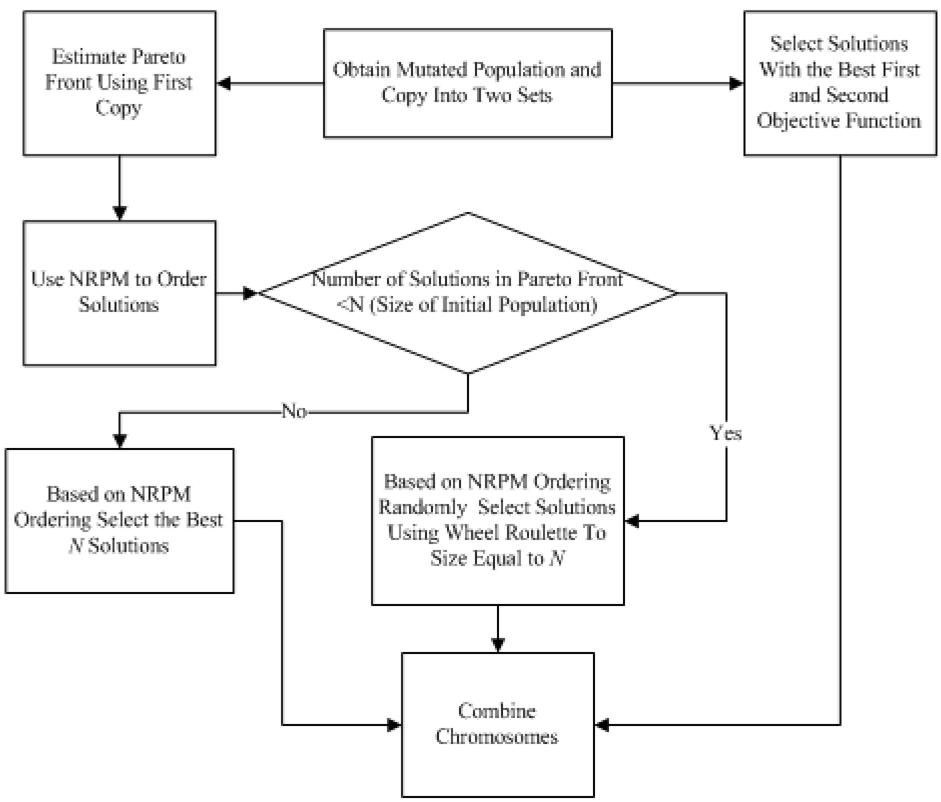

Figure 6.4 Selection Procedure

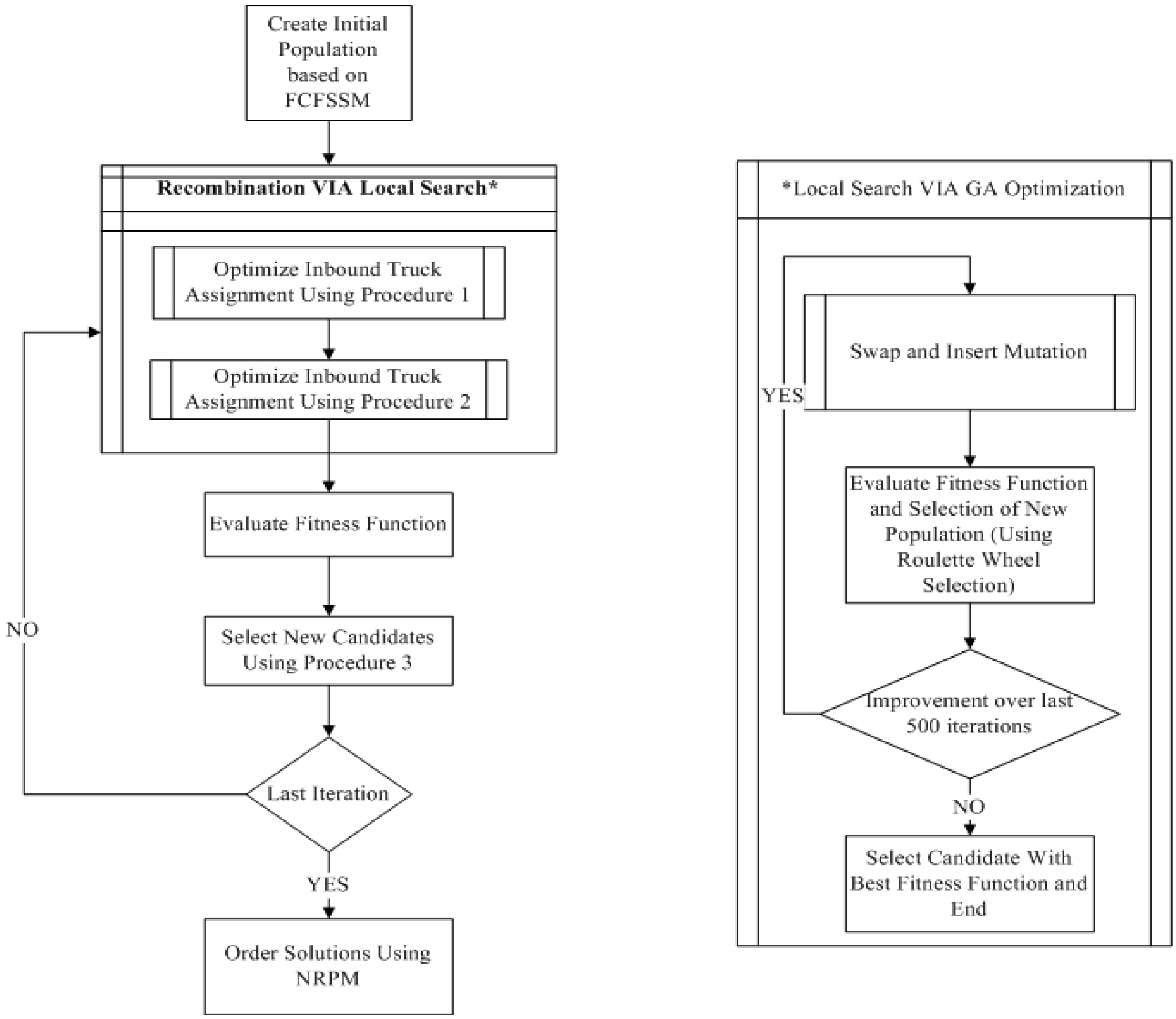

Figure 6.5 MOMA Flowchart

7. CONCLUSIONS AND FUTURE RESEARCH

In this thesis, two models are formulated dealing with the truck scheduling problem (scheduling of ITs and OTs to the available IDs and ODs) at a cross dock facility. In the models it is assumed that the handling time is a variable based on at which door trucks are located, i.e. handling time depends on the travel distance of forklifts from IDs to ODs. In model 1, the service completion time is estimated to check if the departure time window requests are meet. In model 2, the total stay time of the cargo inside the facility is estimated by finding the difference between the service starting time of ITs and OTs to evaluate the temporary storage which could occur. Thus, using the models, trucks were scheduled at the available doors with the objectives to minimize total service time for all trucks, as well as minimize the total storage time for inbound and outbound cargo. To solve the resulting problem, a MOMA based heuristic was constructed. Future research will focus on testing the proposed resolution algorithm using real life test problem instances. Future research will also consider the forklift to truck assignment (inside the facility), which will affect the truck handling time. In the models presented, it was assumed that a sufficient number of forklifts are available so that it is not necessary to consider the forklifts to truck assignment. Thus, forklift assignment does not affect the handling time estimation. In real life, the number of forklifts is limited. An operational model combining the forklift to truck assignment would better estimate the truck handling time and capture real life operations in more detail and accuracy. The third model combining truck-to-door assignment and forklift-to-door assignment is under development.

References

1. Alvarez-Perez, G., J. Gonzalez-Velarde, and J. Fowler. "Crossdocking-Just In Time Scheduling: An Alternative Solution Approach". *Journal of Operational Research Society* Vol.60, No.4 (2009): pp. 554-64.

2. Baker,P. "The Design and Operation of Distribution Centers Within Agile Supply Chains". *International Journal of Production Economics* Vol. 111, No. 1 (2008): pp. 27-41.

3. Bartholdi, J.J. and K. Gue. "Reducing Labor Costs in An LTL Crossdocking Terminal". *Operations Research* Vol.48, No.6 (2000).

4. Bartholdi, J.J. and K. Gue. "The Best Shape for A Crossdock". *Transportation Science* Vol.38, No.2 (2004).

5. Bartholdi, J.J. and S.T. Hackman. 2008. *Warehouse & Distribution Science.* Georgia Institute of Technology, Atlanta, GA 30332-0205 USA.

6. Boilé, M., M.M. Golias, S. Theofanis. "Applications of Genetic Algorithms to Seaside Container Terminal Operations". *Evolutionary Computation*, ISBN 978-953-7619-X-X (2009).

7. Boysen, N. "Truck Scheduling at Zero-inventory Cross Docking Terminals". *Computers and Operations Research*. Vol. 37 (2010): pp. 32-41.

8. Boysen, N. and M. Fliedner. "Cross Dock Scheduling: Classification, Literature Review and Research Agenda". *Omega*. doi:10.1016/j.omega.(2009).

9. Boysen, N., M. Fliedner, A. Scholl. "Scheduling Inbound and Outbound Trucks at Crossdocking Terminals". *OR Spectrum* DOI10.1007/s00291-008-0139-2 (2007).

10. Bozer, Y.A. and H.J. Carlo. "Optimizing Inbound and Outbound Door Assignments in Less-than-truckload Crossdocks". *IIE Transactions* Vol. 40, No. 11 (2008): pp. 1007-18.

11. Chen, R., B. Fan, G. Tang. "Scheduling Problems in Cross Docking. In: Du, D.-Z., Pardalos, P.M. (Eds)". *Lecture Notes in Computer Science, Combinatorial Optimization and Applications* Vol. 5573 (2009): pp. 421-29.

12. Creek, S. Cross docking trends report. Outsourced Logistics 2008. <*http://outsourced-logistics.com/field_reports/cross_docking_trends_report_0808/*>(Accessed on 07/23/2009).

13. Dobrusky, F.G. "Optimal Location of Cross Docking Centers for A Distribution Network in Argentina". *Industry Engineering: Columbia University*, 2003.

14. Donald, R.H, J.V. Vate, M. Zhang. Network Design for Load-driven Cross Docking Systems. *The Logistics Institute, Georgia Institute of Technology*, Atlanta, GA. 1999.

15. Drury, J. and P. Falconer. Building for Industrial Storage and Distribution, *Architectural Press*, Burlington MA, 2003.

16. Eiben, A.E. and J.E. Smith. *Introduction to Evolutionary Computing*, Springer, USA, 2003.

17. Goldberg, D.E. "Genetic Algorithms in Search, Optimization, and Machine-learning". *Reading, MA: Addison-Wesley*, 1989.

18. Golias, M.M., M. Boilé, S. Theofanis, A.H. Taboada. "A Multi-objective Decision and Analysis Approach for the Berth Scheduling Problem". *International Journal of Information Technology Project Management* (in print).

19. Golias, M.M., G.K. Saharidis, S. Ivey, H.E. Haralambides, K. Ji. (2009) "Optimizing Inbound and Outbound Door Operations at A Cross-docking Facility". *International Journal of Production Economics* (Under review).

20. Golias, M.M., G.K.D. Saharidis, M. Boilé, S. Theofanis. (2010) "Bi-objective and a Bi-level Approach for The Scheduling of Inbound Trucks at a Cross-docking Facility". *International Journal of Information Systems and Supply Chain Management.* (Under second review).

21. Golias, M.M., G.K.D. Saharidis, M. Boile, S. Theofanis, T. Zhang. "Scheduling Inbound Trucks at a Cross-docking Facility: A Multi-objective and A Hierarchical Approach". *International Conference on Industrial Engineering and Systems Management, Montreal, Canada* (2009).

22. Gue, K. "The Effects of Trailer Scheduling on the Layout of Freight Terminals". *Transportation Science* Vol. 33, No.4 (1999).

23. Hart, W.E., N. Krasnogor, J.E. Smith. "Memetic Evolutionary Algorithms. In: Hart W.E., Krasnogor N., Smith J.E., (Eds.). Recent Advances in Memetic Algorithms". *Studies in Fuzziness and Soft Computing*, Springer, New York, 166 (2005): pp. 3-27.

24. Katz, J. "Meeting at the crossdock". *Industry Week* vol.255, no.5 (2006): p. 50.

25. Laumar, M. "Facility Logistics: Approaches and Solutions to Next Generation Challenges". *Engineering Management Innovation Series*. Auerbach Publications (2008).

26. Lee, Y.H., J.W. Jung, K.M. Lee. "Vehicle Routing Scheduling for Cross Docking in the Supply Chain". *Comput. Ind. Eng*. Vol. 51, No.2 (2006): pp. 247-56.

27. Li, Y., A. Lim, B. Rodrigues. "Crossdocking-JIT Scheduling With Time Windows". *Journal of the Operational Research Society* Vol. 55 (2004): pp. 1342-51.

28. Miao, Z., A. Lim, H. Ma. "Truck Dock Assignment Problem With Operational Time Constraint Within Crossdocks". *European Journal of Operational Research* Vol. 4031, (2006): pp. 262-71.

29. Moscato, P. 1999. "Memetic Algorithms: A Short Introduction. In: Corne, D., Dorigo, M., Glover, F., (Eds)". *New Ideas in Optimization*. London: McGraw-Hill, 219-234.

30. Nguyen, Q.H., Y.S. Ong, N. Krasnogor. "A Study on the Design Issues of Memetic Algorithm". *IEEE Congress on Evolutionary Computation,* Singapore, ISBN: 978-1-4244-1339-3 (2007): pp. 2390-97.

31. Rouwenhorst, B., B. Reuter, V. Stockrahm, G. J. van Houtum, R. J. Mantel and W. H. M. Zijm. "Warehouse Design and Control: Framework and Literature Review", *EJOR* Vol. 122 (2000) :pgs 515-33.

32. Saxena, R. "Cross Docking Demystified". *Industrial Engineer* (2007).

33. Shakeri, M, M. Low, Z. Li. "A Generic Model for Crossdock Truck Scheduling and Truck-to-Door Assignment Problems". *Industrial Informatics,* (2008). INDIN 2008.6th .

34. Song, K. and F. Chen. "Scheduling Cross Docking Logistics Optimization Problem With Multiple Inbound Vehicles and One Outbound Vehicle. In: Chen F, editor". *Automation and Logistics*, (2007) IEEE International Conference.

35. Sung, C. and S. Song. "Integrated Service Network Design for A Cross Docking Supply Chain Network". *Journal of the Operational Research Society* Vol. 54 (2003): pp. 1283-95.

36. Sunil, C. and P. Meindl. *Supply chain management: Strategy, planning, and operation*, New Jersey: Prentice-Hall, 2002.

37. Tsui, L.Y. and C-H. Chang. "A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards". *Computers & Industrial Engineering* Vol. 19 (1990): pp. 309–12.

38. Tsui, L.Y. and C-H. Chang. "An Optimal Solution to Dock Door Assignment Problem". *Computers & Industrial Engineering* Vol. 23 (1992): pp.283–86.

39. Van den Berg, J.P. and W.H.M. Zijm. "Models for Warehouse Management: Classification and Examples", *International Journal of Production Economics* Vol. 59 (1999):pp.519-28.

40. Wang, J., A. Regan, M. Tsai. "Minimizing Departure Time for Outgoing Trucks in a Crossdock". Proceedings of the Annual Meeting of the *Transportation Research Board* (2008): Paper: #08-0620.

41. Wen, M., J. Larsen, J. Clausen, J-F. Cordeau and G. Laporte. "Vehicle Routing With Cross Docking". *Journal of the Operational Research Society* (2008) doi: 10.1057/jors.2008.108.

42. Wong, W.K. and Y.S.Leung. "Genetic Optimization of Fabric Utilization in Apparel Manufacturing". *International Journal of Production Economics* Vol.114, No. 1 (2008): pp. 376-87.

43. Yeh, W-C. "A Memetic Algorithm for the n/2/Flowshop/αF+βCmax Scheduling Problem". *International Journal of Advanced Manufacturing and Technology* Vol. 20 (2002): pp. 464-73.

44. Yu, W. and PJ. Egbelu. "Scheduling of Inbound and Outbound Trucks in Cross Docking Systems With Temporary Storage". *European Journal of Operational Research* Vol.184 (2006): pp. 377–96.

45. Zäpfel, G. and M.Wasner. "Warehouse Sequencing in the Steel Supply Chain as a Generalized Job Shop Model". *International Journal of Production Economics* Vol. 104, No. 2 (2006): pp. 482-501.