A bi-objective model to minimize service and storage time at a cross dock facility

MIHALIS M. Golias[a,c,1], **STEPHANIE Ivey**[a,b], **KAIFENG Ji**[a,b], **Lipinski M.E.**[a,b]

[a]Department of Civil Engineering, University of Memphis, Memphis, 3815 Central Ave Memphis, TN 38152, USA

[b]Intermodal Freight Transportation Institute, University of Memphis, Memphis, 3815 Central Ave Memphis, TN 38152, USA

[c]Freight and Maritime Program, Center for Advanced Infrastructure and Transportation, Rutgers, The State University of New Jersey, 100 Brett Road, Piscataway, NJ 08854, USA

Abstract: Cross-docking is a relatively new concept of warehousing with limited storage time to maximize the efficiency of transshipment of goods. 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 paper, we deal with the latter type of problem, and propose 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. Our 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.

Keywords: Logistics, Cross docking, Memetic Algorithms, Heuristics, Optimization

---

[1] Corresponding Author: Phone: 901-678-3048, Fax: 901-678-3026,

Emails: mihalisgolias@yahoo.com, ssalyers@memphis.edu, johnnyjijkf@gmail.com, mlipinsk@memphis.edu

INTRODUCTION

In today’s customer driven economy, moving products quickly, efficiently, and cost effectively offers a distinctive comparative advantage to 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. 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 main costs in a supply chain, cross-docking becomes an attractive alternative to warehousing. Cross-dock is a material handling operation, whereby products move quickly and directly from inbound trucks (ITs) to outbound trucks (OTs), after being resorted or consolidated with limited storage time, normally not exceeding 24 hours (Saxena, 2007; Laumar, 2008). Facilities using this type of operation are generally in a “hub-and-spoke” arrangement, where (de)consolidation of cargo occurs, as in the case of transshipment, with products delivered to customers in truckloads (TL). 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 Corporation, 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).

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 (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.

The latter type of problem is considered here. We deal with the scheduling of ITs and OTs to the available IDs and ODs and propose 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. Our 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 our paper we assume 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 next section of model assumptions and formulation). To our knowledge this is the first time in the published literature that such a truck scheduling policy (under these assumptions) has been presented. The mathematical formulation, although linear, 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 multi-objective memetic algorithm is developed and presented.

A limitation of our model 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 paper we assume that a sufficient number of forklifts are available so that trucks do not have to wait while at the IDs and ODs. The remainder of the paper is structured as follows. The next section provides the mathematical formulation of the proposed truck-to-door policy, followed by a description of the resolution algorithm used to solve the resulting problem. The final section concludes the paper and suggests future research directions.

MODEL FORMULATION

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 to the IDs that ITs with cargo for them are served (fig. 1a). 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 (fig. 1b). These conditions do not change even if cargo is temporarily stored within the facility (i.e. two-touch complexity shown in fig. 1c, 1d). 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).

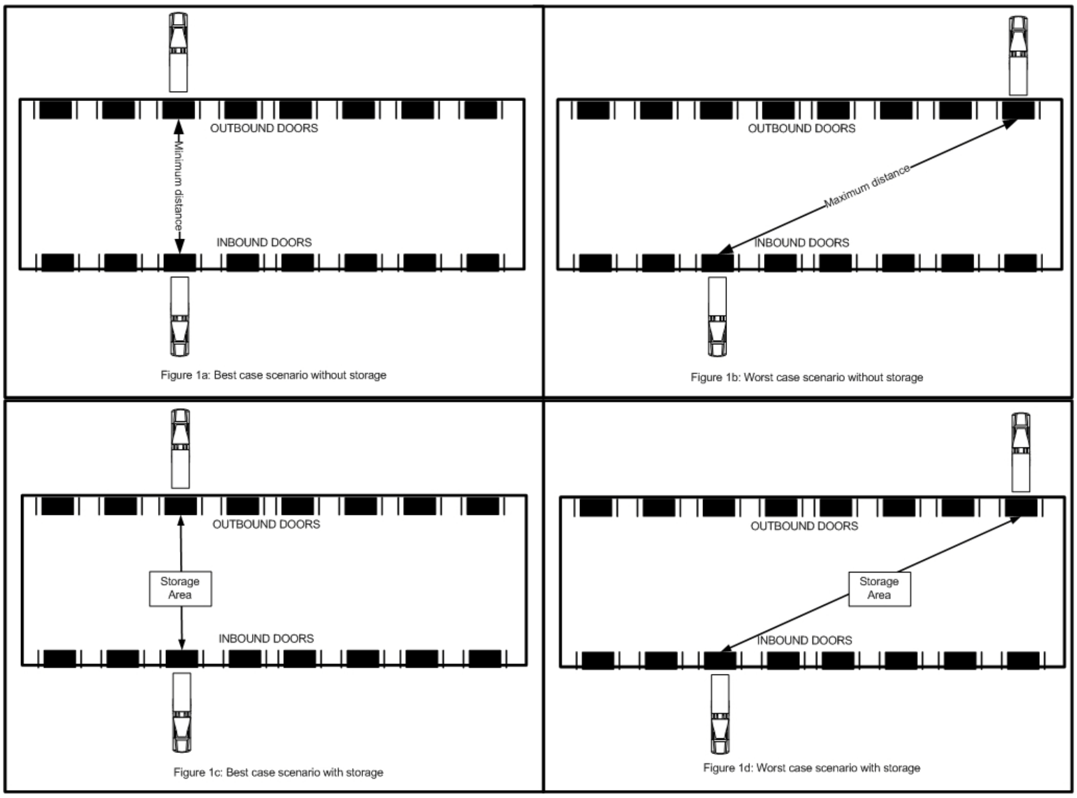


Figure 1a: Best case scenario without storage

Figure 1b: Worst case scenario without storage

Figure 1c: Best case scenario with storage

Figure 1d: Worst case scenario with storage

Figure 1. Best and Worst Case Truck-to-Door Assignments

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, 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

$x_{ij} \in \{0,1\} \forall i \in I_1, I_2, j \in J_1, J_2$	=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 BCDP) minimizing the total service time and total storage time can be formulated as follows:

$$\min\left[\sum_{j\in J_1,J_2}(t_j - A_j) + \sum_{i\in I_1,I_2}\sum_{j\in J_1,J_2} c_j x_{ij}\right] \tag{1.1}$$

$$\min\left[\sum_{a\in J_2}\sum_{b\in J_1} T_{ab}U_{ab}\right] \tag{1.2}$$

Subject To:

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{gathered} c_j \geq \sum_{j^{'}} U_{jj'} K_{jj'} \left(\sum_a \sum_b (F_{ab} x_{aj} + tu) \right) - M(1 - y_{bj'}), \\ \forall a \in I_1, b \in I_2, j \in J_1, j^{'} \in J_2 \end{gathered} \tag{15}$$

$$\Pi_j \geq (t_i + c_i - t_j) K_{ij}, \forall i \in J_1, j \in J_2 \tag{16}$$

$$\begin{gathered} c_j \geq \sum_{j^{'}} U_{jj'} K_{jj'} \left(\sum_a \sum_b (F_{ab} x_{aj'} + tl) \right) - M(1 - y_{bj}) - \Pi_j, \\ \forall a \in I_1, b \in I_2, j \in J_2, j^{'} \in J_1 \end{gathered} \tag{17}$$

$$T_{ab} \geq (t_a - t_b - c_b x_{ib}) K_{ba}, \forall a \in J_2, b \in J_1, i \in I_1 \tag{18}$$

$$y_{ab} = 0, \forall a \in J_1, b \in J_2 \tag{1919}$$

$$y_{ab} = 0, \forall b \in J_1, a \in J_2 \tag{20}$$

$$x_{ab} = 0, \forall a \in I_2, b \in J_1 \tag{211}$$

$$x_{ab} = 0, \forall a \in I_1, b \in J_2 \tag{222}$$

$$x_{ij} \in \{0,1\}, \forall i \in I_1, I_2 \, j \in J_1, J_2 \tag{233}$$

$$y_{ab} \in \{0,1\}, \forall a,b \in J_1, J_2, a \neq b \tag{244}$$

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 paper we assume 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 sets (2) ensure 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.

RESOLUTION ALGORITHM

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 BCDP, considering either or both objectives[ii], 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).

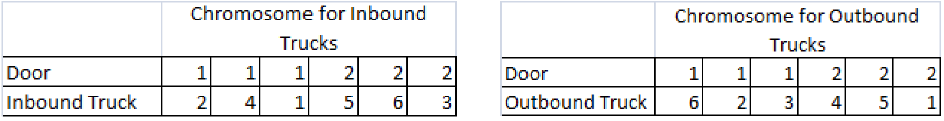
Before continuing with the description of the MA, two definitions by Nguyen et al. (2003), used here, are presented for purposes of consistency:

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.

Chromosomal Representation

In scheduling problems, similar to the one presented here, integer chromosomal representation is more adequate (Eiben and Smith, 2003; Boilé et al. 2009; Wong and Leung, 2008) and is thus adopted. An illustration of the chromosome structure used here is given in figure 2 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 2, 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 (FCFSSQ).



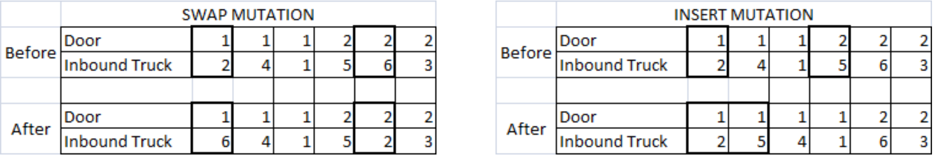
	Chromosome for Inbound Trucks					
Door	1	1	1	2	2	2
Inbound Truck	2	4	1	5	6	3

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

Figure 2. Illustration of Chromosome Representation

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 3 for the same examples used in figure 2. 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 (Boilé et al. 2009).

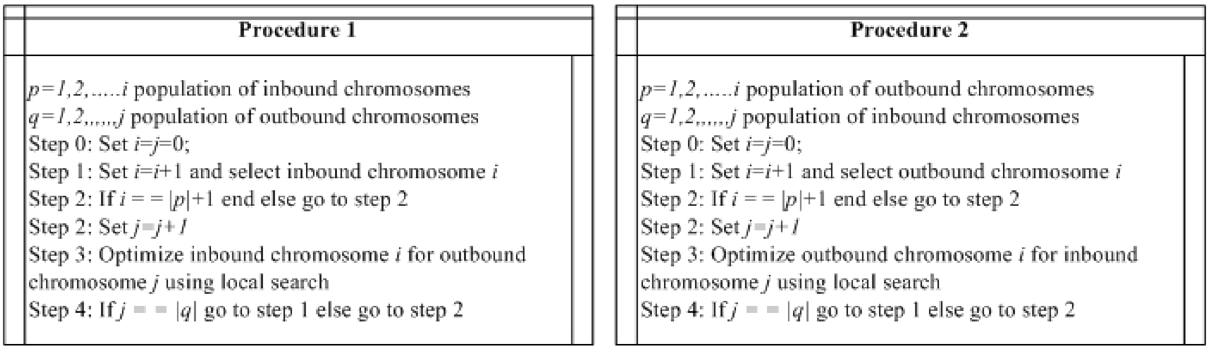


	SWAP MUTATION						
Before	Door	1	1	1	2	2	2
	Inbound Truck	2	4	1	5	6	3
After	Door	1	1	1	2	2	2
	Inbound Truck	6	4	1	5	2	3

	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 3. 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 4.



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 4. 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 5. The proposed MOMA is shown in figure 6 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.

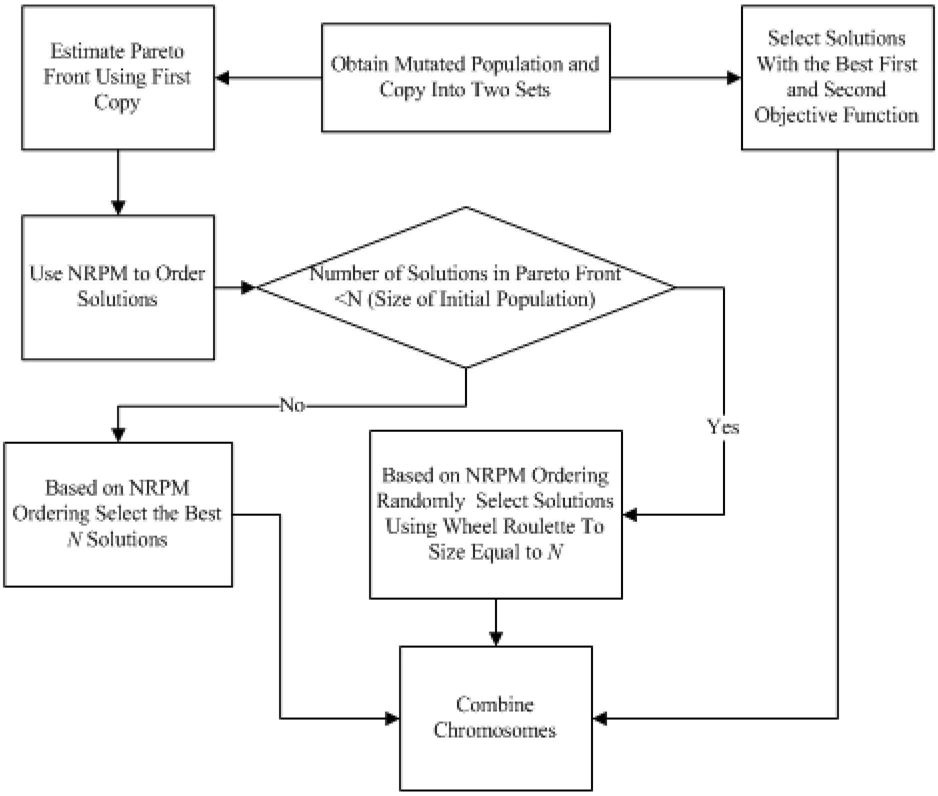


Figure 5. Selection Procedure

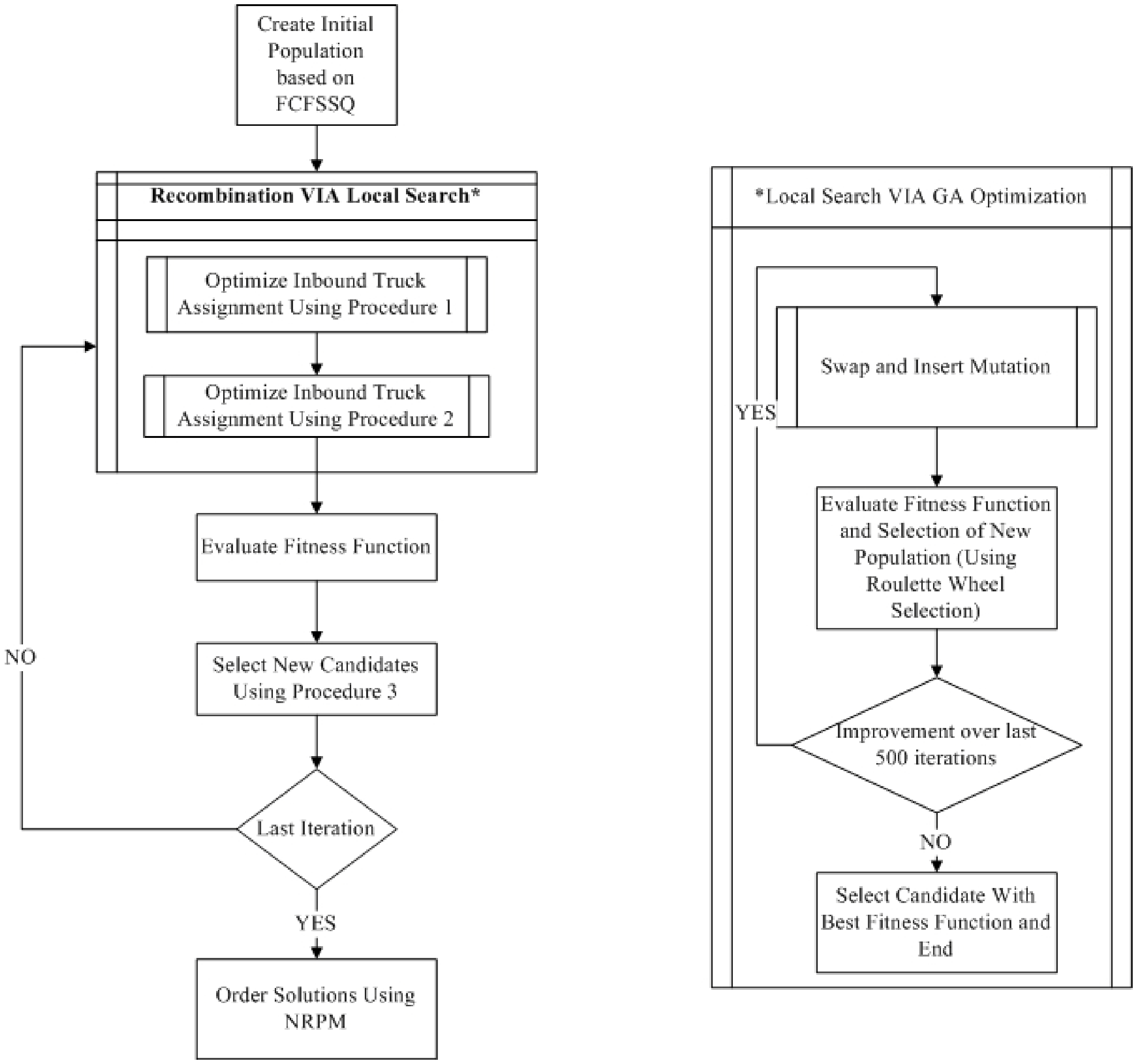


Figure 6. MOMA Flowchart

CONCLUSIONS AND FUTURE RESEARCH

In this paper we formulated the scheduling of ITs and OTs to the available IDs and ODs at a cross dock facility as a flowshop machine scheduling problem. In the model we assumed that the handling time is a variable based on which door trucks are located. This means that the handling time depends on the travel distance of forklifts from IDs to ODs. However, temporary storage may happen during this process and the handling time does not include the stay time of cargo inside the facility. We estimate the total stay time of the cargo inside the facility by finding the difference between the service starting time of ITs and OTs. Thus, under our model, 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 our model, we assumed that a sufficient number of forklifts are available so that we do not need to consider the forklifts to truck assignment. 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.

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. 554-564.
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., Gue, K. Reducing labor costs in an LTL crossdocking terminal. *Operations Research*, Vol.48, No.6, 2000
4. Boilé M., Golias, M.M., Theofanis, S. Applications of genetic algorithms to seaside container terminal operations. *Evolutionary Computation*, ISBN 978-953-7619-X-X, 2009.
5. Boysen N. Truck scheduling at zero-inventory cross docking terminals. *Computers and Operations Research*. Vol. 37, 2010, pp. 32-41
6. Boysen N., Fliedner M. Cross dock scheduling: Classification, literature review and research agenda. *Omega*. doi:10.1016/j.omega.2009.10.008
7. Bozer Y.A., Carlo, H.J. Optimizing inbound and outbound door assignments in less-than-truckload crossdocks. *IIE Transactions*, Vol. 40, No. 11, 2008, pp. 1007-18.
8. Chen, R., Fan, B., Tang, G. 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. 421-429.
9. 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).*
10. Dobrusky, F.G. Optimal location of cross docking centers for a distribution network in Argentina. *Industry Engineering: Columbia University*, 2003.
11. 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.
12. Eiben, A.E., Smith, J.E., 2003. *Introduction to Evolutionary Computing*, Springer, USA.
13. Golias M.M., Boilé M., Theofanis S., Taboada A.H. A multi-objective decision and analysis approach for the berth scheduling problem. *International Journal of Information Technology Project Management* (in print)
14. Golias M.M., Saharidis G.K., Ivey S., Haralambides H.E., Ji K. (2009) Optimizing inbound and outbound door operations at a cross-docking facility. *International Journal of Production Economics* (Under review)
15. Golias M.M., Saharidis G.K.D., Boilé M., Theofanis S. (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)
16. Gue K. The Effects of Trailer Scheduling on the Layout of Freight Terminals. *Transportation Science*, Vol. 33, No.4, 1999
17. Hart, W.E., Krasnogor, N., Smith, J.E. 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, 2008.

18. Laumar M. Facility Logistics: Approaches and solutions to next generation challenges. *Engineering Management Innovation Series.* Auerbach Publications 2008.
19. 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.
20. Li Y., A. Lim, B. Rodrigues. Crossdocking-JIT scheduling with time windows. *Journal of the Operational Research Society*, Vol. 55, 2004, pp. 1342-51.
21. 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.
22. 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.
23. Saxena R. Cross docking demystified. *Industrial Engineer* 2007.
24. Song K., Chen F. 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.
25. 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.
26. Sunil C., P. Meindl. *Supply chain management: Strategy, planning, and operation*, New Jersey: Prentice-Hall, 2002
27. Tsui L.Y., Chang C-H. A microcomputer based decision support tool for assigning dock doors in freight yards. *Computers & Industrial Engineering.* Vol. 19, 1990, pp. 309–12.
28. Tsui L.Y., Chang C-H. An optimal solution to dock door assignment problem. *Computers & Industrial Engineering,* Vol. 23, 1992, pp.283–286.
29. Van den Berg, J.P., Zijm, W.H.M. Models for warehouse management: classification and examples, *International Journal of Production Economics*, Vol. 59, 1999,pp.519-28.
30. Wang J, Regan A, Tsai M. Minimizing Departure Time for Outgoing Trucks in a Crossdock. Proceedings of the Annual Meeting of the *Transportation Research Board,* 2008, Paper: #08-0620.
31. Wen M., J. Larsen, J. Clausen, J-F Cordeau, and G. Laporte G. Vehicle routing with cross docking. *Journal of the Operational Research Society*, 2008, doi: 10.1057/jors.2008.108
32. Wong, W.K., Leung, Y.S., 2008. Genetic optimization of fabric utilization in apparel manufacturing. *International Journal of Production Economics*, Vol.114, No. 1,2008, pp. 376-387
33. Yu W, Egbelu PJ. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, Vol.184, 2006, pp. 377–96.
34. Zäpfel G., Wasner M. 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

---

[i] 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

[ii] Using weights to combine the two objective functions into one