Advances in Truck Scheduling at a Cross Dock Facility

Mihalis M. Golias, Department of Civil Engineering, University of Memphis, Memphis, TN, USA
Georgios K. D. Saharidis, Department of Mechanical Engineering, University of Thessaly, Volos, Greece
Stephanie Ivey, Department of Civil Engineering, University of Memphis, Memphis, TN, USA
Hercules E. Haralambides, Center for Maritime Economics & Logistics (MEL), Faculty of Economics, Erasmus University Rotterdam, Rotterdam, The Netherlands

ABSTRACT

In this paper the authors deal with the scheduling of inbound and outbound trucks to the available inbound and outbound doors at a cross dock facility. They assume that all trucks served at the facility need to meet several deadlines for deliveries and pick-ups and thus request a departure time window from the facility, penalizing the facility operator, on a unit of time basis, if that deadline is not met. To solve the resulting problem with reasonable computational effort, a memetic algorithm is developed and a number of computational examples show the efficiency of the proposed solution algorithm and the advantages of scheduling inbound and outbound trucks simultaneously, as opposed to sequentially.

Keywords: Cross-Docking, Heuristics, Memetic Algorithms, Optimization Methods, Truck Scheduling

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 crossdocking 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 & Meindl, 2002). Products are subsequently sent out to retailers upon request. However, as inventory costs represent one of the main costs in a supply chain, crossdocking 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 needs, normally not exceeding 24 hours (Laumar, 2008). This type of facility is generally

DOI: 10.4018/ijisscm.2013070102
used in “hub-and-spoke” arrangements, 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, showed that 52% of the respondents used cross dock and 13% plan to do so within the next one to two years (Laumar, 2008; Saddle Creek Corporation, 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 & 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., 2009; Song & Chen, 2007; Bozer & Carlo, 2008; Yu & 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 locations, 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. For an excellent critical literature review of cross dock operations we refer to Boysen and Fliedner (2010).

The latter type of problem is considered here. We deal with the scheduling of ITs and OTs to the available IDs and ODs. Our truck scheduling approach builds on two previous papers by Li et al., (2004) and Alvarez-Perez et al., (2009) who presented optimization models to schedule ITs and OTs so as 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. Our model introduces departure requests in the form of a time window, where the facility operator is penalized if that departure time window deadline is not met (as opposed to the point in time departure in Li et al., 2004 and 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 total delayed or early departures affects the productivity of the facility (i.e. total throughput). We consider this aspect by including an addi-
tional term, minimizing total service time for all the trucks. Finally, and unlike the models presented by Li et al. (2004) and Alvarez-Perez et al. (2009), 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 travel time of the forklifts carrying cargo from the ITs to the OTs. Considering the distance forklifts will travel time in the cross dock facility makes the modeling approach more realistic as it accounts for one of the most time consuming operations in a cross-docking scenario (if coordination of IT-to-ID and OT-to-OD is not considered). This assumption and its effect are further discussed in the next section where the model assumptions and formulation are presented.

To our knowledge this is the first time in the published literature that a truck scheduling policy under these assumptions described has been presented. The mathematical formulation, although linear, is still \textit{NP-Hard}, 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 (MA) is developed and the scheduling policy and solution algorithm are evaluated through a number of numerical examples. 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 (for both the ITs and OTs) and by the number of forklifts assigned to a door (or truck) 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 designated 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; impeding the (un)loading and storage operations. The simultaneous scheduling of trucks-to-doors and forklifts-to-trucks is left as future research and in this paper we assume that a sufficient number of forklifts are available so that trucks do not have to wait idle at the IDs or 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 solution algorithm used to solve the resulting problem. Afterwards we evaluate the performance of the proposed solution algorithm and policy, based on a number of numerical examples. The final section concludes the paper and suggests future research directions.

**MODEL ASSUMPTIONS AND 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 (Figure 1a). The handling times of the trucks (ITs, OTs or both) increase with the forklift travel time (Figure 1b). These conditions do not change even if cargo is temporarily stored within the facility (i.e. two-touch complexity shown in Figure 1c, Figure 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). We also assume that each truck (inbound or outbound) makes a time window request for departing the facility. If the truck departs before or after this time window the facility operator is penalized per time unit of departure delay.

In order to formulate the problem of truck scheduling at the available doors under these assumptions, with the objective to minimize the total service time and total cost from tardy and early departures for all the trucks, 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:**
  
  \( x_{ab} \in \{0, 1\}, \forall a \in I_1, I_2, b \in J_1, J_2 \) = 1 if truck \( b \) is served at door \( a \) and zero otherwise

  \( y_{ab} \in \{0, 1\}, \forall a, b \in J_1, J_2 \) = 1 if truck \( b \) is served at the same door as truck \( a \) as its immediate successor and zero otherwise

  \( f_j \in \{0, 1\}, \forall j \in J_1, J_2 \) = 1 if truck \( j \) is served as the first truck (at the door it is assigned) and zero otherwise

  \( l_j \in \{0, 1\}, \forall j \in J_1, J_2 \) = 1 if truck \( j \) is served as the last truck (at the door it is assigned) and zero otherwise
• Parameters:

\[ LR_{j}, j \in J_1, J_2 \text{ latest requested departure time of truck } j \]

\[ ER_{j}, j \in J_1, J_2 \text{ earliest requested departure time of truck } j \]

\[ F_{ab}, a \in I_1, b \in I_2 \text{ moving time of one unit forklift from door } a \text{ to door } b \text{ (in minutes)} \]

\[ U_{ab}, a \in J_1, b \in J_2 \text{ quantity of commodity carried by inbound truck } a \text{ going to truck } b \text{ (in forklift units)} \]

\[ K_{ab}, a \in J_1, b \in J_2 \text{ 1 if incoming truck } a \text{ carries cargo to be shipped out by outgoing truck } b \text{ and zero otherwise} \]

\[ A_{j}, j \in J_1, J_2 \text{ arrival time of truck } j \]

\[ S_i, i \in I_1, I_2 \text{ time door } i \text{ becomes available for the first time in the planning horizon} \]

\[ a_{j}, j \in J_1, J_2 \text{ cost per minute of early departures} \]

\[ b_{j}, j \in J_1, J_2 \text{ cost per minute of tardy departures} \]

\[ tl \text{ loading time for one unit of commodity} \]

\[ tu \text{ unloading time for one unit of commodity} \]

\[ M \text{ large positive number} \]

\[ N_1, N_2 \text{ normalizing factors (positive numbers)} \]

• Auxiliary Variables:

\[ LD_{j} \in R^{+} \cup \{0\}, \forall j \in J_1, J_2 \text{ minutes of late departure of truck } j \]

\[ ED_{j} \in R^{+} \cup \{0\}, \forall j \in J_1, J_2 \text{ minutes of early departure of truck } j \]

\[ t_{j} \in R^{+} \cup \{0\}, \forall j \in J_1, J_2 \text{ start time of service for truck } j \text{ at its assigned door} \]

\[ c_{j} \in R^{+} \cup \{0\}, \forall j \in J_1, J_2 \text{ handling time of truck } j \]

\[ \Pi_{j} \in R^{+} \cup \{0\}, \forall j \in J_1, J_2 \text{ continuous positive variable number estimating the total handling time} \]

A model minimizing the total service time and minimizing the total cost from tardy and early departures (and from now on referred to as CDSP), can be formulated as shown in Box 1.

The objective function minimizes total service time for all trucks (first component), and 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 Appendix C formulations \( SO_1 \) and \( SO_2 \)) given the same feasible space as the original problem. The normalization of the objective function components is necessary as they are in different units (i.e. time and monetary units). Constraint set (2) ensures that each IT and OT is 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 (18) estimate the
Box 1.

\[
\begin{align*}
\min & \left[ \frac{\sum_{i, j_1, j_2} (t_j - A_j) + \sum_{i, j_1, j_2} c_j x_{ij} + \sum_{j_1, j_2} b_j LD_j + \sum_{j_1, j_2} a_j ED_j}{N_1} \right] \\
\text{Subject To:} & \\
\sum_{i, l_1, l_2} x_{ij} &= 1, \forall j \in J_1, J_2 \quad (2) \\
f_a + \sum_{a, l_1, l_2} y_{ab} &= 1, \forall b \in J_1, J_2 \quad (3) \\
l_a + \sum_{b, l_1, l_2} y_{ab} &= 1, \forall a \in J_1, J_2 \quad (4) \\
f_a + f_b &\leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a, b \in J_1, a \neq b \quad (5) \\
l_a + l_b &\leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a, b \in J_1, a \neq b \quad (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 \quad (7) \\
f_a + f_b &\leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a, b \in J_2, a \neq b \quad (8) \\
l_a + l_b &\leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a, b \in J_2, a \neq b \quad (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 \quad (10) \\
t_j &\geq A_j \sum_{i, l_1, l_2} x_{ij}, \forall j \in J_1, J_2 \quad (11) \\
t_j &\geq S_j, \forall j \in J_1, J_2, i \in I_1, I_2 \quad (12) \\
t_j &\geq t_0 + \sum_{i, l_1} c_j x_{ij}, \forall a, b \in J_1, a \neq b \quad (13) \\
t_j &\geq t_0 + \sum_{i, l_2} c_j x_{ij}, \forall a, b \in J_2, a \neq b \quad (14) \\
ED_j &\geq ER_j - t_j - \sum_{a, i, l_1} c_j x_{ij}, \forall j \in J_1 \quad (15) \\
ED_j &\geq ER_j - t_j - \sum_{a, i, l_2} c_j x_{ij}, \forall j \in J_2 \quad (16) \\
LD_j &\geq t_j + \sum_{i, l_1} c_j x_{ij}, \forall j \in J_1 \quad (17) \\
LD_j &\geq t_j + \sum_{i, l_2} c_j x_{ij}, \forall j \in J_2 \quad (18) \\
c_j &\geq \sum_{j} U_{j} K_{j} \left( \sum_{a} \sum_{b} (F_{ab} x_{ij} + t_i) - M(1 - y_{ij}), \forall a \in I_1, b \in I_2, j \in J_1, j \in J_2 \right) \quad (19) \\
\Pi_j &\geq (t_j + c_i - t_j) K_{j}, \forall i \in I, j \in J \quad (20) \\
c_j &\geq \sum_{j} U_{j} K_{j} \left( \sum_{a} \sum_{b} (F_{ab} x_{ij} + t_i) - M(1 - y_{ij}) - \Pi_j, \forall a \in I_1, b \in I_2, j \in J_2, j \in J_1 \right) \quad (21) \\
y_{ij} &\geq 0, \forall a \in I_1, b \in I_2 \quad (22) \\
x_{ij} &\geq 0, \forall a \in I_1, b \in I_2 \quad (23)
\end{align*}
\]
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 OT is equal to the time required to transfer and load all the commodities from the IDs, less the time that the ITs are served before the OT starts service. Constraint sets (22) and (23) ensure that an IT cannot be served at an OD and vice versa.

**SOLUTION ALGORITHM**

The problem formulation presented in the previous section is **NP-Hard**, as it can be reduced to the Multi-Traveling Salesman Problem (MSTP). It is thus highly unlikely that an exact solution algorithm exists that can solve real life instances of the problem in tractable computational times. To overcome this, we create a multi-population MA. 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 a computational limit (set a-priori) is reached (usually the time required—i.e. CPU time— or the number of iterations for the algorithm to converge). We would like to note that although this problem is an operational type of scheduling problem, the CPU time is not very critical, in real world conditions, as long as it remains less than the average handling time of a single truck. This limit is adequate to allow for rescheduling of trucks not yet served without any additional delays. 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. We choose to use MAs as their advantage over EAs for combinatorial problems is consistent especially on complex search spaces (Wu, 2001; Garg, 2009). 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).

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 & Smith, 2003) 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).

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.

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. 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 solved in series, with the same objective function and constraints as the original problem presented previously. The application of the local search increases the computational time.
at each iteration but significantly improves the search of the feasible space and reduces the total number of iterations needed.

As was discussed above each chromosome consists of two separate sub-chromosomes: one for the IT-to-ID assignment and one for the OT-to-OD one. 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: \( f_{l} = 1 \) and, \( t_{u} = 500 \) 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 and selection operators as described here, and insert and swap mutation for recombination.

**Fitness Function and Selection**

Since the problem is a minimization problem, the smaller the objective function value, the higher the fitness value. We use the fitness function proposed by Goldberg (1989). This is given by: \( z_{i}(x) = \max(f_{i}^{o}(x)) - f_{i}^{t}(x) \), where \( f_{i}^{t}(x) \) is the objective function value and \( z_{i}(x) \) 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 where the left part of the flowchart shows the MA and the right side the GA applied as the local search. The algorithm is terminated at 15,000 iterations or when the objective function does not improve for 500 consecutive iterations.

**COMPUTATIONAL EXAMPLES**

We create 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. 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 from the ITs to the OTs was created randomly (i.e. an IT can carry cargo for any OT irrespective of their arrival time difference). An upper limit to the amount of cargo destined from one IT to one OT, is set equal to 30% of the total cargo carried by the IT. The only other restriction we assume is that an IT can carry cargo for a maximum of five OTs. The impact of these assumptions is tested in the previous subsections.

Travel time of a forklift from the IDs to the ODs is estimated based on the geometry of the facility and a constant forklift speed. We assume a cross dock facility of an \( I \) shape, with the IDs and ODs on opposite sides of the facility. Each door (loading and unloading bay) is assumed to have a width of 15ft and the distance between two doors is assumed equal to 8 ft. The width of the facility is taken as equal to 200ft (Figure 5). For more details on the dimensions of a cross dock facility we refer readers to Drury and Falconer (2003). 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 solution algorithm. The same is not
true if congestion effects within the facility are to be included in the model. Modifying both the model and the solution algorithm to meet such assumptions is left to future research.

Four types of experiments were performed on the dataset. The first experiment was geared towards evaluating the influence of the learning frequency and intensity values on the objective function value and CPU time of the proposed MA. The second experiment aimed at evaluating the performance of the proposed solution algorithm. The third type of experiment compares the proposed scheduling policy to a policy that schedules trucks sequentially (i.e. first ITs assuming an average handling time and then the OTs given the ITs assignment). Finally, the last type of experiment looks at: a) the performance of the proposed solution 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.
Sensitivity Analysis of MA’s Parameter

As discussed, the parameters of the learning frequency and intensity were set to $f_{il} = 1$ and $t_{il} = 500$. We evaluate the influence of these parameters on the time required for the MA to converge and on the objective function value. 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 of these combinations, the 30 problem instances were solved and the change in the objective function and CPU time were recorded. The number of chromosomes selected to undergo the local search at each iteration was selected randomly using the roulette wheel selection (35), with a selection probability for each chromosome equal to:

$$P(\text{po}_i | \text{po}) = \frac{f(\text{po}_i)}{\text{max}_i(f(\text{po}_i))}$$

where $P$ is the conditional probability that chromosome $\text{po}_i$ will be selected for the local search at each iteration given the total population $\text{po}$, and $f(\text{po}_i)$ is the value of the objective function of chromosome $\text{po}_i$ at the current iteration. As expected, the objective function value decreased and CPU time increased with the increase in the values of these parameters. The maximum difference in 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 (and on average 300 generations with a maximum of 400 generations and a minimum of 220), while the minimum difference in the objective function value was 10%. For this type
of problem and for practical applications, 10 minutes is considered a small computational expense. Thus, the maximum values for both parameters can be used, so that the best objective function value is obtained.

MA Efficiency

This subsection presents a set of computational examples to evaluate the performance of the proposed MA. As the problem presented here is computational intractable, we were unable to obtain optimal solutions even for small instances of the problem. Thus, to evaluate the performance of the heuristic we compare the convergence CPU time and the objective function values to those obtained from the proposed MA and a variation of an MA proposed by Yeh (2002) (from now on referred to as MA) for the flowshop scheduling problem. Heuristic MA, as with any heuristic not created for the problem at hand, was applied with the following modifications:

1. Initial solutions were obtained using the fcfssq (also used in the ma);
2. Crossover probabilities were decreased and set equal to zero (to avoid infeasibility issues);
3. Mutation probabilities were increased and set equal to 1; and
4. 10 machines were used (instead of 2).

We solved the 30 problem instances (presented in the beginning of this section) using both metaheuristics (we will refer to the schedules obtained by MA and MA as and ) for the flowshop scheduling problem. Heuristic , as with any heuristic not created for the problem at hand, was applied with the following modifications:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( S_1 ) vs ( S_2 )</th>
<th>( S_1 ) vs ( S_2 )</th>
<th>( S_1 ) vs ( S_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5%</td>
<td>17%</td>
<td>17%</td>
</tr>
<tr>
<td>2</td>
<td>6%</td>
<td>9%</td>
<td>15%</td>
</tr>
<tr>
<td>3</td>
<td>15%</td>
<td>9%</td>
<td>4%</td>
</tr>
<tr>
<td>4</td>
<td>11%</td>
<td>5%</td>
<td>9%</td>
</tr>
<tr>
<td>5</td>
<td>7%</td>
<td>8%</td>
<td>12%</td>
</tr>
<tr>
<td>6</td>
<td>5%</td>
<td>12%</td>
<td>7%</td>
</tr>
<tr>
<td>7</td>
<td>8%</td>
<td>9%</td>
<td>18%</td>
</tr>
<tr>
<td>8</td>
<td>8%</td>
<td>9%</td>
<td>12%</td>
</tr>
<tr>
<td>9</td>
<td>4%</td>
<td>9%</td>
<td>12%</td>
</tr>
<tr>
<td>10</td>
<td>6%</td>
<td>9%</td>
<td>14%</td>
</tr>
</tbody>
</table>

\( S_1 \): Schedule obtained from MA (until convergence)
\( S_2 \): Schedule obtained from MA (until convergence)
Table 2. Improvement of CPU time required for full convergence (MA VS MA$_2$)

<table>
<thead>
<tr>
<th>Truck Inter-Arrival</th>
<th>5 min</th>
<th>10 min</th>
<th>15 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>$S_1$ vs $S_2$</td>
<td>$S_1$ vs $S_2$</td>
<td>$S_1$ vs $S_2$</td>
</tr>
<tr>
<td>1</td>
<td>29%</td>
<td>45%</td>
<td>72%</td>
</tr>
<tr>
<td>2</td>
<td>45%</td>
<td>39%</td>
<td>70%</td>
</tr>
<tr>
<td>3</td>
<td>45%</td>
<td>58%</td>
<td>43%</td>
</tr>
<tr>
<td>4</td>
<td>14%</td>
<td>60%</td>
<td>41%</td>
</tr>
<tr>
<td>5</td>
<td>43%</td>
<td>48%</td>
<td>76%</td>
</tr>
<tr>
<td>6</td>
<td>26%</td>
<td>53%</td>
<td>65%</td>
</tr>
<tr>
<td>7</td>
<td>20%</td>
<td>45%</td>
<td>70%</td>
</tr>
<tr>
<td>8</td>
<td>8%</td>
<td>33%</td>
<td>69%</td>
</tr>
<tr>
<td>9</td>
<td>35%</td>
<td>36%</td>
<td>65%</td>
</tr>
<tr>
<td>10</td>
<td>23%</td>
<td>70%</td>
<td>52%</td>
</tr>
</tbody>
</table>

$S_1$: Schedule obtained from MA (until convergence)
$S_2$: Schedule obtained from MA$_2$ (until convergence)

to keep the truck to door ratio to comparable levels to the rest of the datasets. The remaining required data (e.g. truck handling times, cargo association etc) were based on the same assumptions as the initial datasets. These new instances were solved using the proposed MA. For these instances the average number of generations needed for convergence, for the larger instances, did not change significantly (an average of 10% increase was observed as compared to the 10 ID and 10 OD instances). On the other hand the CPU time increased, on average, by 20 minutes, which is still within acceptable limits.

Scheduling Policy Evaluation

The proposed truck-to-door scheduling policy (i.e. combined schedule of ITs and OTs) is next 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 Appendix 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 known 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 ODs furthest and closest to each ID for each IT. For example (as shown in Figure 6) 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: $C_{a3} = \sum_b U_{ab} F_{33}, b \in J_2$ and the time to unload all the cargo to door 10 is equal to: $C_{a3} = \sum_b U_{ab} F_{a10}, b \in J_2$. In this paper we assume an average handling time of:

$$C_{a3} = \sum_b U_{ab} \left( \frac{F_{33} + F_{a10}}{2} \right), a \in I_1, b \in I_2$$
Even with the assumption of known handling times for the ITs (based on Figure 6), solving the sequential problem requires solving two computationally intractable 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 3. The first column of Table

### Table 3. Differences in the objective function between the CS and the SS approach (no restriction)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>5 min</th>
<th>10 min</th>
<th>15 min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inbound</td>
<td>Outbound</td>
<td>Total</td>
</tr>
<tr>
<td>1</td>
<td>23%</td>
<td>8%</td>
<td>15%</td>
</tr>
<tr>
<td>2</td>
<td>24%</td>
<td>11%</td>
<td>16%</td>
</tr>
<tr>
<td>3</td>
<td>32%</td>
<td>12%</td>
<td>21%</td>
</tr>
<tr>
<td>4</td>
<td>30%</td>
<td>12%</td>
<td>19%</td>
</tr>
<tr>
<td>5</td>
<td>29%</td>
<td>3%</td>
<td>14%</td>
</tr>
<tr>
<td>6</td>
<td>10%</td>
<td>2%</td>
<td>5%</td>
</tr>
<tr>
<td>7</td>
<td>22%</td>
<td>8%</td>
<td>14%</td>
</tr>
<tr>
<td>8</td>
<td>28%</td>
<td>11%</td>
<td>17%</td>
</tr>
<tr>
<td>9</td>
<td>17%</td>
<td>5%</td>
<td>10%</td>
</tr>
<tr>
<td>10</td>
<td>18%</td>
<td>2%</td>
<td>8%</td>
</tr>
</tbody>
</table>
3 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 3 show the same results, with truck inter-arrival times of 10 and 15 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 (of the objective function value) of 23%, 8%, and 15% for the ITs, OTs, and for both the ITs and OTs respectively. Looking at the results for the remaining datasets and inter-arrival times, we observe that the proposed scheduling policy always provides an improved schedule when compared to the schedule obtained using the sequential scheduling approach.

**ITs and OTs Arrival Time Association**

In our initial data assumptions we allowed ITs to carry cargo for any OT irrespective of its arrival time. In this subsection we evaluate 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 \( S_2 \). Using the assumptions for the data presented in the beginning of this section we create 10 additional datasets for each inter-arrival time (i.e. 5, 10, and 15 minutes). Unlike the datasets used in the experiments of the previous subsection (where ITs could carry cargo for any OT irrespective 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 a time less than \( t+4 \) hours and \( t+6 \) hours respectively). Results from these experiments are shown in Table 4 and Table 5 for the combined and sequential scheduling models. As can be seen, the benefits remain significant. For these new datasets convergence CPU times improves slightly with the maximum reduction reaching 10% when compared to the convergence CPU times of the examples in the previous subsection (i.e. results reported in Table 3).

**CPU Time and ITs and OTs Cargo Association**

In our initial data assumptions we allowed ITs to carry cargo for a maximum of five OTs. In this subsection we evaluate the effect of this relationship 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 \( S_2 \). In the datasets used for these experiments, ITs were allowed to carry cargo for a maximum of ten and fifteen OTs. Results from these experiments are shown in Table 6 and Table 7 for the combined and sequential scheduling models. One can observe that the benefits remain significant. Unlike results in subsections above, the convergence CPU time for these datasets almost doubles (on average) compared to the convergence CPU time of the examples previously mentioned, reaching a maximum of 20 minutes (for the examples with the five minutes truck inter-arrival time). This increase in computational burden is acceptable as it is less than the average handling time of an average truck (Boysen, 2010).

**CONCLUSION**

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 linear mixed integer problem. We assumed that both ITs and OTs need to meet several deadlines for pick-ups and deliveries and thus request a departure time window from the cross dock facility. Trucks were scheduled at the available doors with the
objective to minimize total service time for all trucks, as well as minimize the total time of delayed and early departures. To solve the resulting problem, an MA based heuristic was constructed. Results from a number of numerical experiments showed that the proposed policy provides improved schedules when compared to policies (where ITs and OTs are scheduled sequentially) that have been proposed to date by the available literature (to our knowledge) as it reduced the total service time for all the trucks by an average of over 20% (over the test instances and different assumptions about the amount of cargo carried by each IT for each OT). We note that although the proposed scheduling policy can be implemented, as is, to any cross dock facility, the proposed solution algorithm is not constructed for use by a cross dock facility.

Table 4. Differences in the objective function between the CS and the SS approach (4 hour restriction)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>5 min</th>
<th>10 min</th>
<th>15 min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inbound</td>
<td>Outbound</td>
<td>Total</td>
</tr>
<tr>
<td>1</td>
<td>15%</td>
<td>35%</td>
<td>13%</td>
</tr>
<tr>
<td>2</td>
<td>24%</td>
<td>34%</td>
<td>17%</td>
</tr>
<tr>
<td>3</td>
<td>17%</td>
<td>39%</td>
<td>9%</td>
</tr>
<tr>
<td>4</td>
<td>25%</td>
<td>33%</td>
<td>14%</td>
</tr>
<tr>
<td>5</td>
<td>24%</td>
<td>35%</td>
<td>17%</td>
</tr>
<tr>
<td>6</td>
<td>18%</td>
<td>49%</td>
<td>13%</td>
</tr>
<tr>
<td>7</td>
<td>20%</td>
<td>30%</td>
<td>13%</td>
</tr>
<tr>
<td>8</td>
<td>24%</td>
<td>33%</td>
<td>12%</td>
</tr>
<tr>
<td>9</td>
<td>26%</td>
<td>46%</td>
<td>13%</td>
</tr>
<tr>
<td>10</td>
<td>30%</td>
<td>33%</td>
<td>19%</td>
</tr>
</tbody>
</table>

Table 5. Differences in the objective function between the CS and the SS approach (6 hour restriction)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>5 min</th>
<th>10 min</th>
<th>15 min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inbound</td>
<td>Outbound</td>
<td>Total</td>
</tr>
<tr>
<td>1</td>
<td>22%</td>
<td>37%</td>
<td>38%</td>
</tr>
<tr>
<td>2</td>
<td>36%</td>
<td>20%</td>
<td>38%</td>
</tr>
<tr>
<td>3</td>
<td>33%</td>
<td>39%</td>
<td>22%</td>
</tr>
<tr>
<td>4</td>
<td>10%</td>
<td>14%</td>
<td>34%</td>
</tr>
<tr>
<td>5</td>
<td>20%</td>
<td>37%</td>
<td>26%</td>
</tr>
<tr>
<td>6</td>
<td>37%</td>
<td>19%</td>
<td>13%</td>
</tr>
<tr>
<td>7</td>
<td>29%</td>
<td>39%</td>
<td>15%</td>
</tr>
<tr>
<td>8</td>
<td>25%</td>
<td>20%</td>
<td>37%</td>
</tr>
<tr>
<td>9</td>
<td>20%</td>
<td>32%</td>
<td>16%</td>
</tr>
<tr>
<td>10</td>
<td>38%</td>
<td>32%</td>
<td>25%</td>
</tr>
</tbody>
</table>
Table 6. Differences in the objective function between the CS and the SS approach (one IT carrying cargo for a maximum of 10 OTs)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Inbound</th>
<th>Outbound</th>
<th>Total</th>
<th>Inbound</th>
<th>Outbound</th>
<th>Total</th>
<th>Inbound</th>
<th>Outbound</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 min</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>24%</td>
<td>11%</td>
<td>17%</td>
<td>37%</td>
<td>22%</td>
<td>26%</td>
<td>22%</td>
<td>9%</td>
<td>13%</td>
</tr>
<tr>
<td>2</td>
<td>18%</td>
<td>6%</td>
<td>11%</td>
<td>35%</td>
<td>17%</td>
<td>24%</td>
<td>37%</td>
<td>14%</td>
<td>21%</td>
</tr>
<tr>
<td>3</td>
<td>15%</td>
<td>3%</td>
<td>9%</td>
<td>46%</td>
<td>27%</td>
<td>36%</td>
<td>32%</td>
<td>21%</td>
<td>24%</td>
</tr>
<tr>
<td>4</td>
<td>26%</td>
<td>10%</td>
<td>17%</td>
<td>27%</td>
<td>7%</td>
<td>14%</td>
<td>42%</td>
<td>27%</td>
<td>32%</td>
</tr>
<tr>
<td>5</td>
<td>30%</td>
<td>16%</td>
<td>21%</td>
<td>23%</td>
<td>14%</td>
<td>17%</td>
<td>53%</td>
<td>28%</td>
<td>38%</td>
</tr>
<tr>
<td>6</td>
<td>28%</td>
<td>10%</td>
<td>16%</td>
<td>35%</td>
<td>21%</td>
<td>26%</td>
<td>41%</td>
<td>22%</td>
<td>29%</td>
</tr>
<tr>
<td>7</td>
<td>27%</td>
<td>10%</td>
<td>16%</td>
<td>30%</td>
<td>12%</td>
<td>18%</td>
<td>41%</td>
<td>18%</td>
<td>26%</td>
</tr>
<tr>
<td>8</td>
<td>30%</td>
<td>10%</td>
<td>18%</td>
<td>38%</td>
<td>19%</td>
<td>26%</td>
<td>31%</td>
<td>14%</td>
<td>19%</td>
</tr>
<tr>
<td>9</td>
<td>22%</td>
<td>6%</td>
<td>12%</td>
<td>28%</td>
<td>15%</td>
<td>20%</td>
<td>38%</td>
<td>18%</td>
<td>26%</td>
</tr>
<tr>
<td>10</td>
<td>19%</td>
<td>12%</td>
<td>16%</td>
<td>45%</td>
<td>22%</td>
<td>30%</td>
<td>12%</td>
<td>8%</td>
<td>9%</td>
</tr>
</tbody>
</table>

Table 7. Differences in the objective function between the CS and the SS approach (one IT carrying cargo for a maximum of 15 OTs)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Inbound</th>
<th>Outbound</th>
<th>Total</th>
<th>Inbound</th>
<th>Outbound</th>
<th>Total</th>
<th>Inbound</th>
<th>Outbound</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 min</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>25%</td>
<td>9%</td>
<td>16%</td>
<td>29%</td>
<td>21%</td>
<td>24%</td>
<td>30%</td>
<td>18%</td>
<td>22%</td>
</tr>
<tr>
<td>2</td>
<td>23%</td>
<td>11%</td>
<td>8%</td>
<td>32%</td>
<td>14%</td>
<td>20%</td>
<td>45%</td>
<td>21%</td>
<td>28%</td>
</tr>
<tr>
<td>3</td>
<td>23%</td>
<td>7%</td>
<td>13%</td>
<td>28%</td>
<td>13%</td>
<td>18%</td>
<td>43%</td>
<td>13%</td>
<td>25%</td>
</tr>
<tr>
<td>4</td>
<td>19%</td>
<td>9%</td>
<td>8%</td>
<td>39%</td>
<td>12%</td>
<td>19%</td>
<td>21%</td>
<td>13%</td>
<td>15%</td>
</tr>
<tr>
<td>5</td>
<td>22%</td>
<td>12%</td>
<td>16%</td>
<td>32%</td>
<td>11%</td>
<td>16%</td>
<td>36%</td>
<td>15%</td>
<td>21%</td>
</tr>
<tr>
<td>6</td>
<td>27%</td>
<td>12%</td>
<td>18%</td>
<td>33%</td>
<td>13%</td>
<td>19%</td>
<td>25%</td>
<td>14%</td>
<td>17%</td>
</tr>
<tr>
<td>7</td>
<td>16%</td>
<td>2%</td>
<td>8%</td>
<td>43%</td>
<td>18%</td>
<td>27%</td>
<td>55%</td>
<td>23%</td>
<td>35%</td>
</tr>
<tr>
<td>8</td>
<td>18%</td>
<td>2%</td>
<td>9%</td>
<td>41%</td>
<td>12%</td>
<td>22%</td>
<td>20%</td>
<td>8%</td>
<td>13%</td>
</tr>
<tr>
<td>9</td>
<td>23%</td>
<td>1%</td>
<td>11%</td>
<td>37%</td>
<td>13%</td>
<td>21%</td>
<td>43%</td>
<td>17%</td>
<td>24%</td>
</tr>
<tr>
<td>10</td>
<td>24%</td>
<td>11%</td>
<td>17%</td>
<td>38%</td>
<td>20%</td>
<td>26%</td>
<td>31%</td>
<td>11%</td>
<td>16%</td>
</tr>
</tbody>
</table>

Future research is focusing on developing a GUI (Graphical User Interface) that will enable and simplify the use of the proposed MA. Future research is also focusing in a multi-objective problem formulation where a variety of schedules with different levels of service for the IDs, ODs and the facility in total, can be obtained. This approach will allow the facility operator to select the most appropriate schedule based on its objectives. Future research could also focus on benchmarking the performance of the proposed MA using existing cross dock facility test-beds schedules based on real life data that at the moment and to the authors’ knowledge, are not publicly available.
ACKNOWLEDGMENT

This material has been partially supported by the Intermodal Freight Transportation Institute, University of Memphis, TN and Kathikas Institute of Research & Technology, Columbus, MO. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Intermodal Freight Transportation Institute and Kathikas Institute of Research & Technology, Columbus, MO.

REFERENCES


ENDNOTES

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
2 From now on this assumption will be referred to as: ITs-to-OTs arrival time association
3 From now on this assumption will be referred to as: ITs-to-OTs cargo association
4 For fast turnaround a minimum of 13ft m bays are recommended with a 15 ft spacing preferable (Drury and Falconer, 2003)

Mihalis M. Golias is an assistant professor at the Department of Civil Engineering. He holds a Diploma of Civil and Environmental Engineering (2001) from the Aristotle University of Thessaloniki, Greece, and a M.S. (2004), and Ph.D. (2007) in Transportation Engineering from Rutgers, The State University of New Jersey. His research focus is in the area of freight transportation.

Georgios K. D. Saharidis received his Bachelor degree in production engineering and management from the Technical University of Crete, Chania, Greece, in 2001 and his MSc and PhD in supply chain management from Ecole Centrale Paris, France in 2002 and 2006, respectively. Dr Saharidis specializes in developing models, optimization methods and decision support systems for large-scale problems. One of his domains of interest is decomposition methods applied in large-scale problems of the supply chain. He is applying optimization models and algorithms to large-scale problems arising in mechanical engineering, transportation, civil engineering, environmental engineering, chemical engineering, industrial engineering and other problem domains of engineering. His research has offered a number of unique solutions approaches that open new opportunities in the area of mathematical programming. He had developed novel acceleration methods for Benders decomposition algorithm and bi-level optimization which have been applied in different engineering problems. Dr. Saharidis works on different research projects such as freight logistics and maritime transportation, port planning and berth scheduling, intermodal terminals and distribution facilities planning and cross-docking scheduling, freight, shipping, and port logistics systems. The focus of his teaching has been on the development and application of optimization models and algorithms to the design, management and operation of transportation systems and operational research.
Stephanie S. Ivey is an Assistant Professor with the Department of Civil Engineering at the University of Memphis. Her primary research interests are in freight modeling, transportation policy, and undergraduate STEM education. She has been a program director for the Herff College of Engineering’s targeted outreach program, Girls Experiencing Engineering, since its inception in 2004, and has served as program faculty in other co-educational outreach programs. Dr. Ivey is the faculty advisor for the student chapter of the Institute of Transportation Engineers at the University, is part of the ITE Transportation Education Council, and serves as the President for the West Tennessee Branch of the American Society of Civil Engineers.

Hercules Haralambides is Professor of Maritime Economics and logistics at the Erasmus School of Economics and Founding Director of the Erasmus Center for Maritime Economics and Logistics (MEL). He is the founder and editor-in-chief of the quarterly Maritime Economics and Logistics, published by Palgrave Macmillan. Prior to joining EUR, Hercules has been teaching at the University of Piraeus, Greece; Cardiff University, UK; and World Maritime University, Sweden. He is the founder of the Special Interest Group on Maritime Transport and Ports of the World Conference of Transport Research (WCTR), as well as one of the three founders (1990) of the International Association of Maritime Economists (IAME). He researches on shipping and ports policy and his work has laid the ground for Europe’s maritime policies over the last twenty years.
APPENDIX A

IT-to-ID Assignment Model:

\[
\min \left[ \sum_{j \in J_1, j \not= d} (t_j - A_j) + \sum_{i \in I_1, j \in J_1} c_{ij} x_{ij} \right] + \sum_{j \in J_1} b_j LD_j + \sum_{j \in J_1} a_j ED_j \\
\left/ \begin{array}{c}
N_1 \\
N_2
\end{array} \right. 
\] (A.1)

Subject to:

\[\sum_{i \in I_1} x_{ij} = 1, \forall j \in J_1 \] (A.2)

\[f_b + \sum_{a \in I_1, a \neq b} y_{ab} = 1, \forall b \in J_1 \] (A.3)

\[l_a + \sum_{b \in I_1, b \neq a} y_{ab} = 1, \forall a \in J_1 \] (A.4)

\[f_a + f_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a, b \in J_1, a \neq b \] (A.5)

\[l_a + l_b \leq 3 - x_{ia} - x_{ib}, \forall i \in I_1, a, b \in J_1, a \neq b \] (A.6)

\[y_{ab} - 1 \leq x_{ia} - x_{ib} \leq 1 - y_{ab}, \forall i \in I_1, a, b \in J_1, a \neq b \] (A.7)

\[t_j \geq A_j, \forall j \in J_1 \] (A.8)

\[t_j \geq S_j f_j, \forall j \in J_1, i \in I_1 \] (A.9)

\[t_b \geq t_a + \sum_{i \in I_1} c_{ia} x_{ia} - M(1 - y_{ab}), \forall a, b \in J_1, a \neq b \] (A.10)

\[ED_j \geq ER_j - t_j - \sum_{i \in I_1} c_{ij} x_{ij}, \forall j \in J_1 \] (A.11)

\[LD_j \geq t_j + \sum_{i \in I_1} c_{ij} x_{ij} - LR_j, j \in J_1 \] (A.12)

\[c_{ij} = \sum_{b \in I_2} U_{jb} \left( \frac{\max(F_{ib}) - \min(F_{ib})}{2} \right), \forall i \in I_1, j \in J_1 \] (A.13)

\[x_{ij} \in \{0,1\}, \forall i \in I_1, j \in J_1 \] (A.14)

\[y_{ab} \in \{0,1\}, \forall a, b \in J_1, a \neq b \] (A.15)

\[f_j, l_j \in \{0,1\}, \forall j \in J_1 \] (A.16)

\[t_j, LD_j, ED_j \in R^+, \forall \in J_1 \] (A.17)
APPENDIX B

OT-to-OD Assignment Model:

\[
\begin{align*}
\min & \quad \left[ \sum_{j \in J_1} (t_j - A_j) + \sum_{i \in I_1, j \in J_2} c_{ij} x_{ij} \right] + \left[ \sum_{j \in J_2} b_j LD_j + \sum_{j \in J_2} a_j ED_j \right] \\
\text{Subject to:} & \\
\sum_{i \in I_2} x_{ij} &= 1, \forall j \in J_2 \\
f_b + \sum_{a \in J_1 \neq b} y_{ab} &= 1, \forall b \in J_2 \\
l_a + \sum_{b \in J_2 \neq a} y_{ab} &= 1, \forall a \in J_2 \\
f_a + f_b &\leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a, b \in J_2, a \neq b \\
l_a + l_b &\leq 3 - x_{ia} - x_{ib}, \forall i \in I_2, a, b \in J_2, a \neq b \\
y_{ab} - 1 &\leq x_{ia} - x_{ib} \leq 1 - y_{ab}, \forall i \in I_2, a, b \in J_2, a \neq b \\
t_j &\geq A_j, \forall j \in J_2 \\
t_j &\geq S_j f_j \forall j \in J_2, i \in I_2 \\
t_b &\geq t_a + \sum_{i \in I_1} c_{ia} x_{ia} - M(1 - y_{ia}), \forall a, b \in J_2, a \neq b \\
ED_j &\geq ER_j - t_j - \sum_{i \in I_1} c_{ij} x_{ij}, \forall j \in J_2 \\
LD_j &\geq t_j + \sum_{i \in I_1} c_{ij} x_{ij} - LR_j, j \in J_2 \\
\Pi_j &\geq (t_j + c_{ij} - t_j) K_{j}, \forall i \in J_1, j \in J_2 \\
c_j &\geq \sum_j U_{ij} K_{ij} \left( \sum_{a} \sum_{b} (F_{ab} x_{ij} + t_l) \right) - M(1 - y_{ij}) - \Pi_j, \forall a \in I_1, b \in J_2, j \in J_2, j' \in J_1 \\
x_{ij} \in \{0,1\}, \forall i \in I_2, j \in J_2 \\
y_{ab} \in \{0,1\}, \forall a, b \in J_2, a \neq b \\
f_j, t_j \in \{0,1\}, \forall j \in J_2 \\
t_j, LD_j, ED_j \in R^+, \forall j \in J_2
\end{align*}
\]

where \( x_{ij}, t_j, i \in I_1, j \in J_1 \) are inputs from the model in Appendix A.
APPENDIX C

Formulations $SO_1$ and $SO_2$:

$SO_1: N_1 = f_1(x)$

where:

$x = \text{arg min} f_1(x)$

s.t.

$(2) - (14), (19) - (23)$

$f_1(x) = \sum_{j \in J_1, d_1} (t_j - A_j) + \sum_{i \in I_1, d_2} \sum_{j \in J_1, d_2} c_j x_{ij}$

$SO_2: N_2 = f_2(x)$

where:

$x = \text{arg min} f_2(x)$

s.t.

$(2) - (23)$

$f_2(x) = \sum_{j \in J_1, d_2} b_j LD_j + \sum_{j \in J_1, d_2} a_j ED_j$