Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty

Dinçer Konur a,*, Mihalis M. Golias b, c

a Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA
b Civil Engineering Department, University of Memphis, Memphis, TN 38152, USA
c Intermodal Freight Transportation Institute, University of Memphis, Memphis, TN 38152, USA

A R T I C L E   I N F O

Article history:
Received 31 May 2012
Accepted 13 May 2013
Available online 21 May 2013

Keywords:
Truck scheduling
Cross-dock
Arrival time uncertainty
Bi-level optimization

A B S T R A C T

This paper studies scheduling of inbound trucks at the inbound doors of a cross-dock facility under truck arrival time uncertainty. Arrival time of an inbound truck is considered to be unknown. In particular, the cross-dock operator only acknowledges the arrival time window of each truck, i.e., the lower and upper bounds of any inbound truck’s arrival time. In absence of any additional information, the cross-dock operator may use three approaches to determine a scheduling strategy: deterministic approach (which assumes expected truck arrival times are equal to their mid-arrival time windows), pessimistic approach (which assumes the worst truck arrivals will be realized), and optimistic approach (which assumes the best truck arrivals will be realized). In this paper, a bi-level optimization problem is formulated for pessimistic and optimistic approaches. We discuss a Genetic Algorithm (GA) to solve the truck-to-door assignments for given truck arrival times, which solves the deterministic approach. Then the GA is modified to solve the bi-level formulations of the pessimistic and the optimistic approaches. Our numerical studies show that an hybrid approach regarding the pessimistic and the optimistic approaches may outperform all of the three approaches in certain cases.

Published by Elsevier Ltd.

1. Introduction and literature review

Cross-docking is a common practice used by companies to reduce warehousing and transportation costs associated with shipments of different items from multiple origins to multiple destinations. In a typical cross-dock facility, products are unloaded from inbound trucks at the inbound doors, sorted, stored and staged, and then, loaded to outbound trucks at outbound doors. Holding times of products at a cross-dock facility typically do not exceed 24 h (Yu & Egbelu, 2008; Alpan, Larbi, & Penz, 2011) and due to consolidation of various shipments to a destination, cross-docking results in fewer shipments in full truckloads per destination. Advantages of cross-docking is then evident from reduced warehousing (inventory holding) and transportation costs. Benefits of practising cross-docking can be maximized by effective planning in the following stages: (i) integrating cross-docking into the distribution network (see, e.g., Bachlaus, Pandey, Mahajan, Shankar, & Tiwari, 2008; Campbell, 2005; Chen, Guo, Lim, & Rodrigues, 2006; Gümüş & Bookbinder, 2004; Jayaraman & Ross, 2003; Kreng & Chen, 2008; Ross & Jayaraman, 2008; Sung & Song, 2003; Sung & Yang, 2008; Syarif, Yun, & Gen, 2002), (ii) designing a successful cross-dock facility layout (Bartholdi & Gue, 2000, 2004; Gue, 1999; Hauser & Chung, 2006; Heragu, Du, Mantel, & Schuur, 2005; Vis & Roodbergen, 2008; Yanchang & Min, 2009), and (iii) efficient control of cross-dock operations. In literature, there exist studies focusing on problems observed in each of the aforementioned stages. While there are successful examples of companies practicing their own cross-docking operations (see, e.g., Forger, 1995; Gue, 1999; Stalk, Evans, & Shulman, 1992; Witt, 1998), it is not uncommon that cross-docking is outsourced. Majority of cross-docking literature, thus, focuses on cross-dock operations within a cross-dock facility independent of location and layout analysis.

As noted by Agustina, Lee, and Pipmani (2010), cross-docking operations include production allocation (i.e., selecting the products that will go through a specific cross-dock facility), dock assignment (i.e., assigning shipment origins to inbound doors and outbound doors to final destinations), vehicle routing with cross-docking, transshipping products from inbound doors to outbound doors, and cross-dock scheduling (i.e., scheduling inbound trucks in inbound doors for unloading or scheduling outbound trucks in outbound doors for reloading). While there are studies focusing on production allocation (see, e.g., Li, Low, & Lim, 2008, 2009), dock assignment (see, e.g., Bermudez & Cole, 2001; Bozer & Carlo, 2008; Ko, Lee, Choi, & Kim, 2008; Lim, Ma, & Miao,
Cross-dock scheduling seeks to find an optimum truck-to-door assignment and the order of service for trucks considering the cross-dock setting. Boysen and Fliedner (2010) discuss cross-dock attributes that are effective in cross-dock operator’s decision on scheduling inbound or outbound trucks. These attributes include but are not limited to door structure, service policies and requirements, truck handling times, and truck arrivals. While objectives considered in scheduling problems are mostly time related such as minimization of total process times, maximum makespan, tardiness or lateness, restrictive assumptions exist on truck arrival times in the cross-docking scheduling literature. It is commonly assumed that all trucks are available at the beginning of each planning horizon, that is, truck arrival times are identical and equal to zero (see, e.g., Lee, Junga, & Lee, 2006; Wen, Larsen, Clausen, Cordeau, & Laporte, 2009), and transshipment with or within cross-docking (see, e.g., Lim, Mao, Rodrigues, & Xu, 2005; Miao, Fu, Fei, & Wang, 2008), cross-dock scheduling has received the most attention.

Window for each truck is known to the cross-dock operator while the truck arrival time distribution is not known. In an attempt to determine scheduling strategies under truck arrival time uncertainty, assuming that truck arrival times are equal to their expected values is reasonable and minimizes the expected value of an objective function. On the other hand, given only the lower and upper bounds on truck arrival times, this approach is not applicable as expected truck arrival times cannot be estimated. A naive approach would be to assume that the expected truck arrival times are equal to mid-arrival time windows. We refer to this approach as the deterministic approach. Deterministic approach assumes that truck arrival time distributions are symmetric; thus, it disregards possible early or late truck arrivals.

In particular, if the cross-dock operator is pessimistic (risk-averse) about truck arrival times, a scheduling strategy that will perform effectively under the worst possible truck-arrival time distributions should be determined. On the other hand, in case the cross-dock operator is optimistic (risk-taking) about truck arrival times, an efficient scheduling strategy for the best possible truck-arrival time distributions should be sought. One of the contributions of this study is to develop a methodology to solve risk-averse and risk-taking cross-dock operator’s scheduling problem using the pessimistic approach and the optimistic approach, respectively. For both approaches, a bi-level optimization problem is formulated to find a scheduling strategy which minimizes the cross-dock operator’s expected total service time (i.e., handling and expected waiting times of the inbound trucks). Under the pessimistic approach, truck arrival time distributions are assumed to be controlled to minimize the expected total waiting times of any selected scheduling strategy, whereas the truck arrival time distributions are assumed to be controlled to minimize expected total waiting time of any selected scheduling strategy under the optimistic approach.

While the deterministic approach ignores possibility of early or late truck arrivals, the pessimistic and optimistic approaches only represent two extreme approaches to the cross-dock operator’s scheduling problem. Therefore, we propose an alternative, so called risk-neutral, approach to the cross-dock operator’s scheduling problem. Particularly, the alternative approach, which is referred to as hybrid approach, develops scheduling strategies regarding the expected truck arrival times obtained from the pessimistic and optimistic approaches. To solve the scheduling problem under the four different approaches, we develop two Genetic Algorithm based heuristic methods and conduct a set of numerical studies. Our numerical studies show that the hybrid approach is an efficient alternative to the deterministic approach for developing scheduling strategies under truck arrival time uncertainty.

The rest of the paper is organized as follows. Section 2 defines the problem setting and formulates the cross-dock operator’s problem for different approaches. In Section 3, we explain the solution methodology. Section 4 summarizes the result of extensive numerical studies and compare the alternative approaches. We conclude in Section 5 with a brief summary, contributing remarks, and potential future research directions.

2. Problem definition and formulations of different approaches

Consider a cross-dock facility with a set of m inbound doors (ID), which are designated to serve a set of n inbound trucks (IT). Let IDs be indexed by i such that i ∈ {1, 2, ..., m} denotes the set of IDs. Similarly, let ITs be indexed by j such that j ∈ {1, 2, ..., n} denotes the set of ITs. The handling time of an IT consists of the unloading time at the ID and the travel time of the unloading equipment from the ID to the staging area or to the outbound doors. We consider that each IT requires a distinct handling
time depending on the ID it is assigned to. As noted by Boysen et al. (2010), truck-specific handling times reflect practical characteristics of cross-dock operations better. Particularly, we define $c_{ij}$ as the handling time of IT $j$ at ID $i$.

It is assumed that truck arrival times are not known to the cross-dock operator but some acknowledges their lower and upper bounds, i.e., the time window of each IT’s arrival. In particular, let $A_j$ denote the arrival time realization of IT $j$, then $A_j \in [A_j^l, A_j^u]$ where $A_j^l$ is the lower bound of $A_j$ and $A_j^u$ is the upper bound of $A_j$. The objective of the cross-dock operator is to determine a schedule, i.e., an IT-to-ID assignment with service order of the ITs at the IDs, that minimizes the total service time of the ITs at the IDs. Total service times include the total handling time of the ITs and the waiting times of the ITs to be served.\footnote{Note that the objective function considers the time dimension of the problem and ignores the idle time of an ID. When an objective function based on costs is considered, cross-dock operator’s total cost would include the handling costs and the costs associated with truck waiting times. Specifically, waiting time of a truck can be charged to the cross-dock operator regarding the Just-In-Time requirements of the ITs. In such a setting, idle time costs can be ignored as they will not contribute to the total cost of a schedule. On the other hand, the modeling framework presented here can be generalized to include ID idle times.}

Prior to formulating the different approaches to the scheduling problem, we next define the notation used in this section, as well as throughout the paper. Additional notation will be defined as needed.

\begin{align*}
i & \text{ index for IDs, } i \in I = \{1,2,\ldots,m\} \\
j & \text{ index for ITs, } j \in J = \{1,2,\ldots,n\} \\
c_{ij} & \text{ handling time of IT } j \text{ at ID } i \\
A_j^l & \text{ lower bound on IT } j \text{’s arrival time} \\
A_j^u & \text{ upper bound on IT } j \text{’s arrival time} \\
A_j & \text{ IT } j \text{’s arrival time, } A_j \in [A_j^l, A_j^u] \\
t_j & \text{ service start time of IT } j \\
x_{ij} & \text{ variable from IT } j \text{ is served at ID } i, x_{ij} = 1 \text{ if IT } j \text{ is served ID } i, x_{ij} = 0 \text{ otherwise} \\
y_{ab} & \text{ variable from IT at ID } j \text{ to IT at ID } i, y_{ab} = 1 \text{ if IT } b \text{ is assigned to IT } a \text{’s predecessor at the same ID, } y_{ab} = 0 \text{ otherwise} \\
f_j & \text{ f}_j = 1 \text{ if IT } j \text{ is served as the first IT at the ID at it is assigned to, } f_j = 0 \text{ otherwise} \\
l_j & \text{ l}_j = 1 \text{ if IT } j \text{ is served as the last IT at the ID it is assigned to, } l_j = 0 \text{ otherwise} \\
\end{align*}

\[ F(X,A) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} (t_j - A_j), \]  
\[ \text{subject to: } \]
\[ \sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \]
\[ f_j + \sum_{a \in J \setminus j} y_{aj} = 1, \quad \forall j \in J \]
\[ l_j + \sum_{a \in J \setminus j} y_{ja} = 1, \quad \forall j \in J \]
\[ f_a + f_b = 3 - x_{ab} - x_{ba}, \quad \forall i \in I, \forall a,b \in J, a \neq b \]
\[ x_{ab} - 1 \leq x_{ab} - x_{ba} \leq 1 - y_{ab}, \quad \forall i \in I, \forall a,b \in J, a \neq b \]
\[ t_j \geq \sum_{a \in J \setminus j} t_a y_{aj} + \sum_{a \in J \setminus j} c_a x_{a} y_{aj}, \quad \forall j \in J \]
\[ t_j \geq A_j, \quad \forall j \in J \]
\[ x_{ij} \in \{0,1\}, \quad \forall i \in I, \forall j \in J, y_{ab} \in \{0,1\}, \forall a,b \in J, a \neq b \]

The cross-dock operator’s objective in \textbf{SP} is minimization of the total service time. Constraint set defined in Eq. (2) ensures that each IT is only assigned to one ID. While constraints defined in Eq. (4) guarantee that each IT will be either served as the last IT at the ID it is assigned to or preceded by another IT, constraints defined in Eq. (3) restrict ITs to be either served as the first IT at the ID it is assigned to or followed by another IT. Constraint sets defined by Eqs. (5)-(7) ensure that only one IT is served as a first and last IT at each ID. Constraints given in Eqs. (8) and (9) define the service start times of the ITs. In particular, $t_j = \max\{A_j, \sum_{a \in J \setminus j} t_a y_{aj} + \sum_{i \in I} c_i x_{i} y_{ai}, \}$ where $\sum_{a \in J \setminus j} t_a y_{aj} + \sum_{i \in I} c_i x_{i} y_{ai}$ defines the finish time of the IT preceding IT $j$. Eq. (10) is the set of binary constraints.

### 2.2. Formulation of different approaches to the scheduling problem

\textbf{SP} assumes that $A_j$ values are known with certainty. On the other hand, as discussed in Section 1, truck arrival times cannot be estimated due to inherent variabilities in travel times. It is reasonable to solve the scheduling problem under truck arrival time uncertainty assuming that $A_j = A_j^l$ $\forall j \in J$, where $A_j^l$ is the expected arrival time of IT $j$, as this will minimize the expected total service times. Nevertheless, when the only information available is a time window for an IT’s arrival, i.e., $A_j \in [A_j^l, A_j^u]$, $A_j^l$ cannot be specified.

Therefore, we consider three approaches to the scheduling problem varying in their assumptions on estimating $A_j$ values.

Before proceeding with the formulation details of each approach, we note that the truck arrival times are assumed to obey a triangular distribution with unknown mode values to the cross-dock operator. That is, the cross-dock operator still does not know the expected truck arrival times. A triangular truck arrival time distribution has the following probability density function:

\[ f(A) = \frac{2(A - A_l)}{(A_u - A_l)(A_m - A_l)} \text{ for } A_l < A_m < A_u, \]
\[ f(A) = \frac{2(A_u - A)}{(A_u - A_l)(A_u - A_m)} \text{ for } A_l < A < A_m, \]
\[ f(A) = \frac{2(A - A_u)}{(A_u - A_l)(A_u - A_m)} \text{ for } A_m < A < A_u, \]

where $A_l$, $A_m$, and $A_u$ denote the lower, middle, and upper bounds of the triangular distribution. Then one concludes that $E[A] = \frac{A_l + A_m + A_u}{3}$. The rationale behind assuming a triangularly distributed truck arrival times are twofold. Firstly, it can be used to represent asymmetric probability distributions over a defined range in a mathematically tractable way. Second, as it will be discussed, it maintains the continuity of the truck arrival time probability distribution while solving for the pessimistic and opti-
mistic approaches. Next, we explain the deterministic approach, give the bi-level optimization problem formulations of the pessimistic and optimistic approaches, and define an hybrid approach.

2.2.1. Deterministic approach

A naive approach when truck arrival time distribution characteristics are unknown would be to assume that the truck arrival times follow a symmetric distribution, that is, the expected truck arrival times are equal to their mid-arrival time windows. Under this assumption, the cross dock operator can solve SP using these expected truck arrival times. Formally, the deterministic approach can be defined as follows.

Definition 1. $X^D$ is the deterministic schedule that solves SP given that

$$ A_j = \frac{A_j^u + A_j^l}{2} \quad \forall j \in J. $$

However, deterministic schedule disregards possible early or late truck arrivals. Therefore, to capture different characteristics of a cross-dock operator, we next consider pessimistic and optimistic approaches.

2.2.2. Pessimistic approach

Under pessimistic approach, it is assumed that the cross-dock operator is risk-averse about truck arrival time distributions and expects that truck arrivals will maximize the expected waiting times. In particular, this approach can be formulated as a bi-level optimization problem where the upper level problem is the cross-dock operator’s expected total service time minimization problem, i.e., SP given that $A_j = EA_j \quad \forall j \in J$, and the lower level problem is maximization of the expected total waiting times by determining the expected arrival times (assuming that truck arrival times are continuous within a defined window). Note that a bi-level optimization problem corresponds to a two-player sequential game. In particular, we formulate the pessimistic approach as a game-against-demon (GAD). Games of this form assume a strict competition with conflicting objectives (see, e.g., Bell, 2000: Bell & Cassir, 2002). In the setting of truck scheduling problem under consideration, GAD has two players: the cross-dock operator and a so-called demon. The former player is the leader of the game (upper level player) and s/he determines a schedule based solely on an estimation of the expected truck arrival times as it is the best s/he can do to minimize total service time under uncertainty of A_j’s. On the other hand, the demon is the follower (lower level player), and, acknowledging the leader’s IT-to-ID assignment with orders of service, it controls EA_j’s so that the expected total waiting time is maximized. The motivation behind choosing such an approach to the cross-dock operator’s scheduling problem is to help the cross-dock operator find a schedule that, when implemented, performs efficiently for the worst case IT arrivals.

Observe that the upper level optimization problem is given by SP such that $A_j = EA_j \quad \forall j \in J$. EA_j values are determined by the lower level player. Considering triangularly distributed truck arrival times, the lower level player’s decision variables can be defined to be $A_j^m$ values. Then one can determine $EA_j = A_j^u - A_j^m$. Furthermore, finding the EA_j values maximizing expected total waiting times is equivalent to finding the $A_j^m$ values maximizing the total waiting times. Recall that one reason why we assumed triangularly distributed truck arrival times is to maintain the continuity of the truck arrival time distributions while solving the pessimistic and optimistic approaches. In both bi-level optimization problems formulated for each approach, the lower level player controls the mode of each truck’s triangular arrival time distribution, thereby, it controls the expected truck arrival times. This only allows the lower level player to determine the expected truck arrival times such that truck arrival time probability distributions are continuous in the given range. However, if the lower level player is assumed to control expected truck arrival times without any restriction, it may be the case that expected arrival times are equal to the lower or upper bound of the arrival time window, which indicates that probability of an IT arriving at the lower or upper bound of its arrival time window is equal to one while the probability of the arrival times within the arrival time window is zero, i.e., a discontinuous probability distribution within the given time window. The lower level player’s problem in the pessimistic approach then reads as follows for any given $X$.

\[
\text{(LLP)} \quad \max_x H(X, A) = \sum_{j \in J} (t_j - A_j^m)
\]

\[
s.t. \; A_j^m \leq A_j^u, \; \forall j \quad (11)
\]

\[
\sum_{a \in J \cap a_j} t_a y_{aj} + \sum_{i \in J \cap i_j} c_{ia} x_{aj} - A_j^m \leq M(1 - p_j), \; \forall j \quad (12)
\]

\[
A_j^m - \sum_{a \in J \cap a_j} t_a y_{aj} - \sum_{i \in J \cap i_j} c_{ia} x_{aj} \leq M p_j, \; \forall j \quad (13)
\]

\[
\sum_{a \in J \cap a_j} t_a y_{aj} + \sum_{i \in J \cap i_j} c_{ia} x_{aj} + M p_j \geq t_j, \; \forall j \quad (14)
\]

\[
A_j^m + M(1 - p_j) \geq t_j, \; \forall j \quad (15)
\]

\[
p_j \in [0, 1], \quad \forall j \quad (16)
\]

where $A^m$ denotes the vector of $A_j^m$ values. The lower level player’s objective maximizes the total waiting times. Constraints defined in Eq. (11) force mode of IT arrival time distributions to be within the given range (upper and lower bounds). Constraint sets given in Eqs. (13)–(16) estimate the ITs’ expected start times for the lower level player. Given an IT-to-ID assignment, the finish time of IT’s predecessor can be either smaller or greater than the upper bound of IT j’s arrival time. In the latter case, $p_j$ becomes equal to 0; Eqs. (13) and (14) are feasible while Eq. (15) becomes unrestrictive (i.e., $t_j \leq M$) and Eq. (16) sets IT j’s start time equal to the finish time of its predecessor. In the former case, $p_j$ becomes equal to 1; Eqs. (13) and (14) are still feasible while Eq. (16) becomes unrestrictive (i.e., $t_j \leq M$) and Eq. (15) sets IT j’s start time earlier than the upper bound of its arrival time.

For notational simplicity, let $G_l(A)$ denote the feasible region of SP defined by Eqs. (2)–(10) for a given A, i.e., set of feasible solutions of the upper level player. Similarly, let $G_u(X)$ denote the feasible region of LLP defined by Eqs. (11)–(16) for a given X. Then we have the following definition.

Definition 2. Let $(X^*, AP^*)$ be the solution of the following bi-level optimization problem which represents the cross-dock operator’s pessimistic scheduling problem

\[
\text{(P-SP)} \quad \min_x F(X, EA)
\]

\[
s.t. \; X \in G_l(EA)
\]

\[
\max_{A^m} H(X, A^m)
\]

\[
s.t. \; A^m \in G_u(X),
\]

where $EA$ is the vector of $EA_j$ values. $X^*$ is the cross-dock operator’s pessimistic schedule.

Next, we define the optimistic approach to the scheduling problem.

2.2.3. Optimistic approach

The optimistic approach to the scheduling problem is formulated similar to the pessimistic approach. The only difference is in the LLP’s objective, which is now minimization. That is, the lower level player, acknowledging the cross-dock operator’s IT-to-ID
assignment controls $EA_i$s so that the expected total waiting time is minimized. We note that the optimistic approach can be modeled as a single level optimization problem (Tsoukalas, Wiesemann, & Rustem, 2009) as

$$\min_{X} F(X, EA) \quad \text{s.t. Eqs. (2)}-(16).$$

However, in the next definition, we formulate the optimistic approach as a bi-level optimization problem, of which solution is equal to the above single level formulation. The reason behind this is that the above formulation constitutes a mixed integer nonlinear programming problem, and bi-level formulation enables use of an iterative process where only integer programming and linear programming problems are analyzed in each iteration.

**Definition 3.** Let $(X^O, AO^m)$ be the solution of the following bi-level optimization problem which represents the cross-dock operator’s optimistic scheduling problem

$$\min_{X} F(X, EA) \quad \text{s.t. X} \in G_1(EA), \quad \max_{A^m} H(X, A^m) \quad \text{s.t.} \quad A^m \in G_2(X),$$

$X^O$ is the cross-dock operator’s optimistic schedule.

In what follows, we define a hybrid approach alternative to the deterministic, pessimistic, and optimistic approaches.

2.2.4. Hybrid approach

In the hybrid approach, the cross-dock operator determines a schedule assuming that the expected truck arrival times are the arithmetic average of the expected truck arrival times given by the pessimistic and optimistic approaches. That is, the cross-dock operator assumes that the worst and best cases are equally likely. Therefore, the cross-dock operator is neither risk-averse nor risk-taking. The hybrid approach is defined as follows.

**Definition 4.** $X^H$ is the hybrid schedule that solves $SP$ given that

$$A_j = \frac{1}{2} \left[ \left( \frac{A_j^I + A_j^O + AP^m}{3} \right) + \left( \frac{A_j^O + A_j^w + AO^m}{3} \right) \right]$$

$$= \frac{A_j^I + A_j^O + AP^m + AO^m}{6}, \quad \forall j \in J.$$

The following section focuses on solution algorithms developed for each distinct approach.

3. Solution analysis

The first focus of this section is to solve $SP$ given any $A$. Note that $SP$ is an NP-hard problem as machine scheduling problem is a special case of $SP$. Therefore, we propose a Genetic Algorithm (GA) based heuristic to solve $SP$, which implies that we can solve the scheduling problem under the deterministic approach, i.e., $X^I$. Then, we modify the GA to solve $P-SP$ and $O-SP$ to determine the schedules under pessimistic and optimistic approaches, i.e., $X^O$ and $X^H$, respectively. Finally, we explain how $X^H$ is determined.

3.1. Genetic algorithm for IT-to-ID assignment

In this section, we explain the GA heuristic to solve $SP$ for any given $A$. The proposed GA heuristic consists of three stages: (i) chromosomal representation, (ii) chromosomal mutation, (iii) fitness function evaluation and selection process.

Because of the characteristics of a scheduling problem, integer chromosomal representation has been discussed to be more effective compared to the classical binary representation as binary representation can obscure the nature of the search (Eiben & Smith, 2003). For the problem of interest, therefore, we adopt the integer chromosomal representation. An illustration of the chromosome is given in Table 1 for a small problem instance with six ITs and two IDs. As seen in Table 1, the chromosome for the ITs has two rows of six cells (equal to the total number of ITs). The cells in the upper row denote the ID assignment while the lower rows represent an IT and its order of service. For example, IT 5 will be served first at ID 2.

(i) Two different types of mutation are applied as part of the genetic operations at each generation: swap and insert. Swap mutation exchanges two selected ITs slots (i.e., ID and IT is assigned to and its order at this ID). Insert mutation takes an IT and assigns it a slot before or after another IT’s slot. Swap and Insert are illustrated in Tables 2 and 3, respectively, for the problem instance given in Table 1.

(ii) The objective function value of each chromosome is used to evaluate the fitness function value of each chromosome. The chromosome with the minimum total service time is selected as the parent of the next generation.

Algorithm 1, stated below, demonstrates the main steps of the GA heuristic.

**Algorithm 1. Genetic Algorithm for SP**

1. Initialize a set of chromosomes by randomly selecting an IT-to-ID assignment with orders of service.
2. Determine the total service time for each chromosome.
3. Check whether the algorithm converges:
   (a) If it converges, select the chromosome with the minimum total service time as the solution.
   (b) Else, let the chromosome with the minimum total service time be the parent chromosome.
4. Use swap and insert mutations on the parent to generate new set of chromosomes. Go to Step 2.

We have considered two convergence conditions for Algorithm 1: GA terminates either when there is no improvement in the fitness function within 500 consecutive generations or the execution time reaches 1800 s. At each iteration, 20 distinct chromosomes

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Illustration of chromosome representation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT-to-ID Assignment</td>
<td>Chromosome</td>
</tr>
<tr>
<td>ID 1</td>
<td>IT 2</td>
</tr>
<tr>
<td>ID 2</td>
<td>IT 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Illustration of swap mutation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>After</td>
</tr>
<tr>
<td>ID 1</td>
<td>1</td>
</tr>
<tr>
<td>IT 2</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Illustration of insert mutation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>After</td>
</tr>
<tr>
<td>ID 1</td>
<td>1</td>
</tr>
<tr>
<td>IT 2</td>
<td>4</td>
</tr>
</tbody>
</table>
are analyzed; 10 of these chromosomes are generated using insert mutation on the parent and the other 10 chromosomes are generated using swap mutation.

3.2. Solving the bi-level optimization problems

To find a solution to the bi-level optimization problems defined in P-SP and O-SP, we propose another GA based heuristic defined similar to Algorithm 1. Prior to describing the heuristic method, we next provide a brief background on bi-level optimization problems and their solutions.

3.2.1. Background

Bi-level programming problems are the most studied class of hierarchical optimization where a set of optimization problems are solved sequentially (Saharidis & Ierapetritou, 2009). As it is the case in P-SP and O-SP, bi-level optimization problems consist of two nested optimization levels: upper and lower levels. The feasibility of a solution to the upper level optimization problem is defined by the constraints imposed on the upper level along with the lower level optimization problem. Bi-level optimization problems are generally non-convex (Colson, Marcotte, & Savard, 2007) and, even the simplest case, when both the upper and lower level problems are linear, is shown to be NP-hard (Hansen, Jaumard, & Savard, 1992).

Research on hierarchical optimization and its applications mostly focus on bi-level linear, non-linear, and mixed-integer programming. As noted by Saharidis and Ierapetritou (2009), while enumeration methods, which are mostly extreme point algorithms that are pioneered by Bard (1983) and Bialas and Karwan (1984), are used for bi-level linear problems; reformulation methods such as reducing the problem to a single level via use of Karush–Kuhn–Tucker optimality conditions are generally used for bi-level non-linear problems. Nevertheless, despite its applicability, there exists limited research on bi-level mixed-integer problems. We refer the reader to Dempe (2003) and Colson, Marcotte, and Savard (2005) for further discussion on bi-level optimization problems and their solution methods.

Generally, a bi-level optimization framework fixes an upper level’s feasible solution in the lower level problem; and, the pair consisting of this solution and the corresponding optimum solution of the lower level, defines a feasible solution to the bi-level problem. The set of such solutions is referred to as inducible region. Hence, it is a reasonable approach to use an interactive scheme, where each iteration, for a given solution of the upper level, evaluates this solution by considering the reaction of the lower level. We use this interactive approach in the following GA based heuristic proposed next.

3.2.2. Genetic algorithm for the pessimistic and optimistic approaches

The GA based heuristic proposed for the bi-level formulations of the pessimistic and optimistic approaches follows similar steps with Algorithm 1. The difference is in the objective function evaluation. To evaluate the objective function value of a given IT-to-ID assignment with orders of service, one needs to determine the worst possible truck arrival times for the pessimistic approach and the best possible truck arrival times for the optimistic approach, i.e., one needs to solve the lower level problem. Recall that, given IT-to-ID assignments with orders of service, the lower level problem is a linear program, and we use CPLEX to solve it. The solution of the lower level problem gives us the mode of the truck arrival time distribution for each IT, and, this mode is then used to calculate the expected total service time of the upper level problem. This expected total service time defines the fitness function value of a schedule.

Algorithm 2, stated below, demonstrates the main steps of the modified GA heuristic used to determine a schedule under pessimistic and optimistic approaches.

Algorithm 2. Modified Genetic Algorithm for P-SP and O-SP:

1. Initialize a set of chromosomes by randomly selecting an IT-to-ID assignment.
2. Evaluate each chromosome:
   (a) If the pessimistic approach is used, let $A_{P}\in \arg\min\{H(X,A_{P})\}$ and the worst case expected total service time of $X$ be $F(X,A_{P})$ such that $EA_{p}=(A_{p}+AP_{p}+A)p_{j}/3 \forall j \in J$.
   (b) If the optimistic approach is used, let $A_{O}\in \arg\min\{H(X,A_{O})\}$ and the best case expected total service time of $X$ be $F(X,A_{O})$ such that $EA_{o}=(A_{o}+AO_{o}+A_{o})/3 \forall j \in J$.
3. Check whether the algorithm converges:
   (a) If it converges, select the chromosome with the minimum expected total service time as the solution.
   (b) Else, let the chromosome with the minimum expected total service time be the parent chromosome.
4. Use swap and insert mutations on the parent to generate new set of chromosomes. Go to Step 2.

For Algorithm 2, 1800 s limit on the execution time is used for convergence. Similar to Algorithm 1, at each iteration, 20 distinct chromosomes, generated using insert and swap mutations, are analyzed. Note that, at termination, $A_{P}$ and $A_{O}$ values are determined by Algorithm 2.

Finally, the solution to the hybrid approach is then achieved by using Algorithm 1 with $A_{j}=\frac{A_{p}A_{o}}{A_{p}+A_{o}}+\frac{AP_{j}AO_{j}}{6} \forall j \in J$. Next, we discuss our numerical analysis.

4. Numerical studies

In this section, we explain our numerical studies that demonstrate the potential savings a cross-dock operator may achieve by using the proposed solution approaches to the scheduling problem under inbound truck arrival time uncertainty. We consider a cross-dock facility with 10 inbound doors and 300 inbound trucks that operates in a 12 h (720 min) shift.

We focus on nine problem sets, where each problem set corresponds to a combination of $C \sim U[15,30], U[30,60], U[45,90]$ and $A_{j} \sim U[0.60,0.90]$ where $C$ denotes the $m \times n$-matrix of $c_{ij}$ values in minutes and $U[u,l]$ denotes a uniform distribution with $u$ and $l$ as the lower and upper limits, respectively. Table 4 gives the details of each problem set.10 problem instances are generated such that $C$ and $A_{j}$ values are randomly selected. In particular, the arrival time window for IT $j$, $[A_{j}^{l},A_{j}^{u}]$, is generated according to a uniform distribution.

Table 4

<table>
<thead>
<tr>
<th>Problem set</th>
<th>$C$</th>
<th>Time window length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U[15,30]</td>
<td>U[0.60]</td>
</tr>
<tr>
<td>2</td>
<td>U[15,30]</td>
<td>U[0.90]</td>
</tr>
<tr>
<td>3</td>
<td>U[15,30]</td>
<td>U[0.120]</td>
</tr>
<tr>
<td>4</td>
<td>U[30,60]</td>
<td>U[0.60]</td>
</tr>
<tr>
<td>5</td>
<td>U[30,60]</td>
<td>U[0.90]</td>
</tr>
<tr>
<td>6</td>
<td>U[30,60]</td>
<td>U[0.120]</td>
</tr>
<tr>
<td>7</td>
<td>U[45,90]</td>
<td>U[0.60]</td>
</tr>
<tr>
<td>8</td>
<td>U[45,90]</td>
<td>U[0.90]</td>
</tr>
<tr>
<td>9</td>
<td>U[45,90]</td>
<td>U[0.120]</td>
</tr>
</tbody>
</table>
as follows. We first generate a random time value for each IT uniformly from within the 12 h shift, i.e., [0, 720]. This time value represents the mid-time window. Then, we randomly generate a time window length for each truck such that \( A_l^u - A_l^l \sim \{0, 60\}, \{0, 90\}, \{0, 120\} \) depending on the problem set the instance is generated (if \( A_l^l < 0 \) or \( A_l^u > 720 \), we adjust the time window such that IT \( J \)'s arrival time window is within [0, 720]).

Each problem instance generated is solved using deterministic, pessimistic, optimistic, and hybrid approaches. In particular, Algorithm 1 is first used to determine \( X^D \). Then, Algorithm 2 is used to determine \( X^P \) and \( X^O \) as well as \( A^P_m \) and \( A^O_m \). Finally, using \( A^P_m \) and \( A^O_m \), Algorithm 1 is executed to determine \( X^H \). After scheduling strategies under each approach are determined for each problem instance, we perform a simulation considering different arrival scenarios to evaluate the performance of these scheduling strategies. We first define the following three possibilities for IT arrivals.

- **Early arrival**: IT \( J \) is more likely to arrive early and in this case \( A_l^P = A_l^u \) and \( EA_l = \frac{2A_l^u + A_l^l}{3} \forall j \in J \).
- **On-time arrival**: IT \( J \) is equally likely to arrive early and late and in this case \( A_l^O = \frac{A_l^u + A_l^l}{2} \) and \( EA_l = \frac{A_l^u + A_l^l}{2} \forall j \in J \).
- **Late arrival**: IT \( J \) is more likely to arrive late and in this case \( A_l^H = A_l^l \) and \( EA_l = \frac{2A_l^u + A_l^l}{3} \forall j \in J \).

We study 15 arrival scenarios such that each scenario has different percentages of early, on-time, and late inbound truck arrivals as shown in Table 5 below.

For a given arrival scenario of a given problem instance, we generate 100 cases where the ITs being early, on-time, and late are randomly selected such that the percentages defined by the arrival scenario are satisfied. Furthermore, for each of these 100 cases, we randomly generate 100 truck arrivals. Truck arrival times are generated using a triangular distribution with \( A_l^u \) and \( A_l^l \) as \( A^u \) and \( A^l \). Finally, using \( A^u \) and \( A^l \), Algorithm 1 is executed to determine \( X^H \).

### Table 5
Arrival scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Early (%)</th>
<th>On-time (%)</th>
<th>Late (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>75</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>75</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>50</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>50</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Scenario 6</td>
<td>50</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>Scenario 7</td>
<td>25</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>Scenario 8</td>
<td>25</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>Scenario 9</td>
<td>25</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>Scenario 10</td>
<td>25</td>
<td>0</td>
<td>75</td>
</tr>
<tr>
<td>Scenario 11</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Scenario 12</td>
<td>0</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>Scenario 13</td>
<td>0</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Scenario 14</td>
<td>0</td>
<td>25</td>
<td>75</td>
</tr>
<tr>
<td>Scenario 15</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table 6
Average total service time over all arrival scenarios for each problem set.

<table>
<thead>
<tr>
<th>Problem set</th>
<th>Scheduling approach</th>
<th>Minimum</th>
<th>Best schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deterministic</td>
<td>Pessimistic</td>
<td>Optimistic</td>
</tr>
<tr>
<td>1</td>
<td>17.291</td>
<td>17.315</td>
<td>17.262</td>
</tr>
<tr>
<td>2</td>
<td>18.176</td>
<td>18.353</td>
<td>18.171</td>
</tr>
<tr>
<td>3</td>
<td>22.527</td>
<td>22.619</td>
<td>22.517</td>
</tr>
<tr>
<td>4</td>
<td>29.664</td>
<td>29.594</td>
<td>29.562</td>
</tr>
<tr>
<td>5</td>
<td>33.555</td>
<td>33.846</td>
<td>33.749</td>
</tr>
<tr>
<td>6</td>
<td>32.938</td>
<td>33.200</td>
<td>32.926</td>
</tr>
<tr>
<td>7</td>
<td>35.331</td>
<td>35.565</td>
<td>35.498</td>
</tr>
<tr>
<td>8</td>
<td>43.558</td>
<td>43.678</td>
<td>43.336</td>
</tr>
<tr>
<td>9</td>
<td>48.837</td>
<td>46.857</td>
<td>46.776</td>
</tr>
<tr>
<td>Average</td>
<td>31.465</td>
<td>31.559</td>
<td>31.422</td>
</tr>
</tbody>
</table>

### Table 7
Average total handling time over all arrival scenarios for each problem set.

<table>
<thead>
<tr>
<th>Problem set</th>
<th>Scheduling approach</th>
<th>Minimum</th>
<th>Best schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deterministic</td>
<td>Pessimistic</td>
<td>Optimistic</td>
</tr>
<tr>
<td>1</td>
<td>9560</td>
<td>9562</td>
<td>9535</td>
</tr>
<tr>
<td>2</td>
<td>9589</td>
<td>9561</td>
<td>9589</td>
</tr>
<tr>
<td>3</td>
<td>9556</td>
<td>9554</td>
<td>9591</td>
</tr>
<tr>
<td>4</td>
<td>10,773</td>
<td>10,770</td>
<td>10,772</td>
</tr>
<tr>
<td>5</td>
<td>10,667</td>
<td>10,664</td>
<td>10,656</td>
</tr>
<tr>
<td>6</td>
<td>10,736</td>
<td>10,774</td>
<td>10,729</td>
</tr>
<tr>
<td>7</td>
<td>11,734</td>
<td>11,745</td>
<td>11,734</td>
</tr>
<tr>
<td>8</td>
<td>11,900</td>
<td>11,919</td>
<td>11,888</td>
</tr>
<tr>
<td>9</td>
<td>11,824</td>
<td>11,829</td>
<td>11,819</td>
</tr>
<tr>
<td>Average</td>
<td>10,704</td>
<td>10,719</td>
<td>10,703</td>
</tr>
</tbody>
</table>
Step 2. For each case, generate 100 truck arrival scenarios, such that each IT's arrival time is randomly selected from the triangular distribution defined by the case.

Tables 6–8 document the average total service time, handling time, and waiting time over all arrival scenarios for each problem set defined in Table 4, respectively.
The following points are observed based on Tables 6–8.

- On average over all problem sets, the scheduling strategy under the hybrid approach results in the minimum total service, handling, and waiting times.
- The scheduling strategy under the optimistic approach outperforms the pessimistic and deterministic approaches on average total service times.
- While the hybrid and the optimistic approaches result in dominant scheduling strategies in terms of total service, handling, and waiting times on average, there is one problem set where the deterministic approach is better in terms of total service time.

It can be concluded from Tables 6–8 that the hybrid approach is an alternative to the deterministic approach, which can result in scheduling strategies that are better in terms of total service, waiting, and handling times. One of the main motivations in this study was to develop an approach alternative to the deterministic approach for the scheduling problem with uncertain truck arrival times. The results observed in Tables 6–8 conclude that the hybrid approach constitutes such an alternative.

Furthermore, in Tables 9 and 10, we document the average total service time and waiting time over all problem sets for each arrival scenario defined in Table 5, respectively. It can be observed that, on average over all problem sets, the scheduling strategy under the hybrid approach results in the minimum total service and waiting times for each arrival scenario. In addition, average total handling times under deterministic, pessimistic, optimistic, and hybrid approaches are 10,704, 10,719, 10,703, and 10,675, respectively. That is, the hybrid approach outperforms the other alternatives in terms of handling times as well.

5. Conclusions and future research

In this paper, we study scheduling of inbound trucks to inbound doors at a cross-dock facility. Our motivation was to provide complete analyses on possible approaches to the scheduling problem in the case of unknown truck arrival times.

We first discussed the deterministic approach, a very naive approach to the scheduling problem under consideration in this paper, which assumes that expected truck arrival times are equal to their mid-time window values. To solve the scheduling problem under deterministic approach, a single-level optimization problem is formulated to determine the IT-to-ID assignments with orders of service minimizing the total expected service time given the expected truck arrival times. In particular, we proposed a Genetic Algorithms based heuristic method to solve IT-to-ID assignments with orders of service given the expected truck arrival times. Our main contribution is in discussing alternative approaches to the deterministic approach for the scheduling problem of interest in this paper.

We analyzed three alternative approaches to the truck scheduling with uncertainty of truck arrival times. The first approach is the pessimistic approach, which represents a risk-averse cross-dock operator. A bi-level optimization problem is formulated to solve the pessimistic approach. The second approach is the optimistic approach, which represents a risk-taking cross-dock operator. Similarly, we formulated a bi-level optimization problem to solve the optimistic approach. A modified Genetic Algorithms based heuristic method is discussed to solve the bi-level optimization problems of the pessimistic and optimistic approaches. Finally, the third approach is the hybrid approach, which considers the worst-case and best-case truck arrival time distributions and gives scheduling strategies accordingly.

In our numerical studies, we compared the performance of the different approaches. Our numerical studies imply that the hybrid and the optimistic approaches outperform the deterministic and the pessimistic approaches for different problem classes. Furthermore, it is observed that the scheduling strategies planned under the hybrid approach dominates the scheduling strategies planned under the other approaches for all of the different truck arrival scenarios considered. These observations suggest that the hybrid approach can be used as an efficient alternative to the deterministic approach for scheduling inbound trucks to inbound doors with truck arrival time uncertainty.

To the best of our knowledge, this paper is the first study that explicitly takes truck arrival time uncertainty into account in different cross-dock scheduling strategies. The tools provided in this study can be used in scheduling problem for different cross-dock operator characteristics. Future research directions would be to study robust scheduling strategies in case of truck arrival time uncertainties. Determining a scheduling strategy with low worst-case and best-case performance difference, i.e., a reliable scheduling strategy, is important for operations planning in cross-dock facilities. One further research question would be to analyze dynamic scheduling with truck arrival time uncertainty.

References


Footnotes:
1 Note that since the IT-to-ID assignment does not change with arrival scenarios, each arrival scenario has the same average total handling times.
2 Note that since the IT-to-ID assignment does not change with arrival scenarios, each arrival scenario has the same average total handling times.


