International Journal of
Operations Research

International Journal of Operations Research Vol. 14, No. 3, 77−87 (2017)

An Ant Colony Optimization for the Multi-Dock Truck Scheduling Problem with Cross-Docking

Chi-Yuan Luo, and Ching-Jung Ting[1]

Department of Industrial Engineering and Management, Yuan Ze University, 135 Yuan-Tung Road, Taoyuan 32003, Taiwan

Abstract: Cross-docking operation is a new distribution strategy for synchronizing inbound and outbound trucks at the terminal. Products move directly from inbound dock to shipping dock without being stored in the distribution center. In this paper, we consider the truck scheduling problem which simultaneously determines dock assignment and truck scheduling of both inbound and outbound trucks for a multi-door cross-docking operation. The objective is to minimize total holding cost at the cross-docking terminal. A mixed integer programming model is first formulated for the problem. Since both dock assignment and truck scheduling problems are NP-hard, this truck scheduling problem is more difficult to solve. Thus we propose an ant colony optimization (ACO) algorithm for the problem. To evaluate the proposed ACO, 24 instances are generated and tested. The computational results and comparison with Gurobi optimizer solutions show that the ACO is competitive.
Keyword —Cross-docking, Truck Scheduling, Dock Assignment, Ant Colony Optimization.

1. INTRODUCTION

In a traditional warehouse, five major operations are performed: receiving, sorting, storing, retrieving, shipping. Products are usually stored and retrieved when customer's order arrives at the warehouse. Cross-docking is a logistics process, which moves merchandise from the receiving dock to shipping dock for shipping without placing it first into storage locations (Material Handling Institute). Therefore the storage and retrieval cost, the two most expensive among those five warehousing operations, are removed. Fig. 1 shows the flow of products in typical cross docking terminal. In a cross-docking system, inbound trucks are assigned to dock doors when they arrive at the cross-docking terminal. Then cargos are unloaded, sorted, moved and loaded onto outbound trucks assigned to the corresponding dock doors, with little or no storage in between. In this way, economies of transportation are realized and customer service level is maintained (Apte and Viswanathan, 2000).

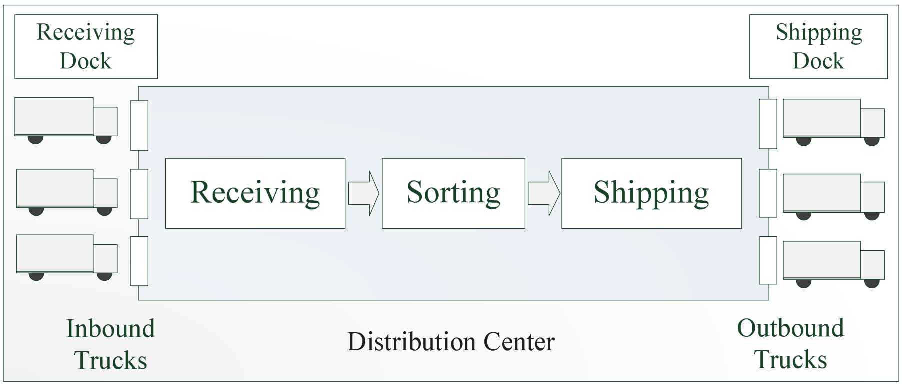

Fig. 1. Typical flow in a cross-docking system.

A recent trend is to adopt cross-docking techniques for efficient supply chain operations (Napolitano, 2000). It can be efficiently controlled to reduce the lead time and the inventory cost. A 2010 survey of 219 logistics professionals conducted by Saddle Creeks (2011) showed that more than two thirds (68.5%) of respondents currently cross dock and 15.1% plan to begin in the next 18 to 24 months. Wal-Mart (Stalk et al., 1992) and Toyota (Witt, 1998) are two well-known examples that reported the successful implementation of cross-docking. Wal-Mart

*Corresponding author's e-mail: ietingcj@saturn.yzu.edu.tw

transport goods through its cross-docking network and reduced its costs of sales by 2-3%. In order to obtain the benefits of cross-docking, many strategic, tactical, or operational optimization problems should be addressed. Dock assignment problem is one of the important optimization problems for daily operations at the cross-dock terminal.

Cross-docking has attracted many researchers' attention in recent years. To the best of our knowledge, several papers (Agustina et al., 2010; Stephan and Boysen, 2011; Van Belle et al., 2012; Buijs et al., 2014; Ladier and Aplan, 2016) present a review of cross-docking research and provide future research opportunities. Agustina et al. (2010) provided a general picture of the mathematical models used in cross-docking planning. In order to efficiently process the transshipment at the CDC, Stephan and Boysen (2011) stated that both inbound and outbound schedules should be synchronized and listed several procedures related to cross-docking. Van Belle et al. (2012) presented an extensive overview of the cross-docking concept and described several characteristics. Buijs et al. (2014) reviewed the cross-docking system operations which include local and network related issues. The authors also mentioned that synchronization between local and network scheduling is important but rare research discussed. One of the key decisions in cross-docking operations is the dock assignment for both inbound and outbound trucks. Ladier and Aplan (2016) proposed a common framework to compare the literature review with on-field observation and platform managers' interviews on cross-docking operations. Future research directions in relation to industrial needs were provided.

The main objective of the cross dock assignment and scheduling problem is to find a good dock allocation to reduce dock delays and travel distance within the cross-docking facility. Boysen and Fliedner (2010), Shuib and Fatthi (2012), and Walha et al. (2014) provide scheduling and dock assignment models for cross-docking operations. Boysen and Fliedner (2010) structured a classification scheme for the cross-docking scheduling problems. Shuib and Fatthi (2012) reviewed the mathematical models for dock door assignment for the daily operation planning, while Walha et al. (2014) defined different types of uncertainties in the cross-docking operations and focused on the dock-door assignment problems.

Tsui and Chang (1990) introduced a bilinear model for determining the inbound and outbound dock allocation, where the objective is to minimize the travel distance in between. The problem is a special case of the quadratic assignment problem (QAP), which is a NP complete problem (Garey and Johnson, 1979). They extended the study by proposing a solution for the bilinear programming model using branch and bound algorithm (Tsui and Chang, 1992). Bozer and Carlo (2008) developed a simulated annealing algorithm to determine the inbound and outbound trailer-to-door assignments in crossdocks without taking into account the congestion.

Oh et al. (2006) considered the mail distribution center in which the different doors are clustered into groups. A non-linear mathematical model was developed with the objective of minimizing the internal travel distance and a three phase heuristic and genetic algorithm were proposed to solve the problem. Yu and Egbelu (2008) addressed a truck scheduling problem where the product assignments from inbound trucks to outbound trucks are determined simultaneously with the docking sequences of the inbound and outbound trucks. They developed three different approaches to solve the problem. Cohen and Keren (2009) analyzed the previous approaches for assigning docks to trucks. They stated that the problem is usually represented by bilinear programming formulation, and introduced a new approach to solve small size problem.

Miao et al. (2009) assumed that the trucks are loaded or unloaded during a fixed time window and the capacity of the crossdock is limited. They formulated the dock assignment problem as an integer programming model with the objective to minimize the operational cost of cargo shipments and penalty cost of unfulfilled shipments and designed tabu search and genetic algorithm for its solution. Miao et al. (2014) dealt with a similar problem, but now each dock door is either exclusively assigned to inbound or outbound trucks. They proposed an adaptive tabu search algorithm to optimize the problem. Li et al. (2004) proposed the multi dock assignment problem as a two-stage parallel machine scheduling problem. They designed and implemented two genetic algorithms (GA) to solve it. The object is minimized the total penalty of earliness cost and tardiness cost. Chen and Song (2009) followed this idea to formulate a mixed integer programming problem with the objective to minimize makespan within warehouse and develop four heuristic.

Liao et al. (2012) developed two differential evolution algorithms (DE) and designed two different methods to compute the makespan. Liao et al. (2013) proposed a dock assignment and sequencing of inbound trucks for a multi-door cross docking under a fixed outbound truck departure schedule. The objective to minimize total weighted tardiness and solved by six different meta heuristic algorithms, which include simulated annealing (SA), tabu search (TS), ant colony optimization algorithm (ACO), DE, and two hybrid DE. Two of best algorithms are ACO and hybrid DE 2, and ACO takes less computational time than hybrid DE 2.Thus ACO can be declared the best among all the six algorithms tested. Yu (2015) tackled the truck scheduling problem with multiple inbound and outbound docks. A mathematical model that extends Yu and Egbelu's (2008) was developed and two heuristic approaches were proposed to solve the problem.

Our study is motivated by the real operations for e-commerce providers in Taiwan. The products collected from a supplier are dedicated to a specific customer. Thus, the inbound truck content is known and all products on a specific inbound truck are dedicated to specific outbound trucks. We focus on inbound truck scheduling and in considering sequencing and dock assignment simultaneously rather than sequencing or dock assignment only. From the supply chain perspective, we would like to reduce the stay time at the terminal for all products and minimize the inventory holding cost at the terminal. Thus the objective of our paper minimize the time in the terminal for all products which is differs from previous studies (Yu and Egbelu, 2008; Boloori Arabani et al., 2011; Liao et al., 2013; Yu, 2015), that minimize the makespan. The main contributions of this study include (i) formulating the truck scheduling problem for a multi-door cross-docking system, (ii) proposing ACO algorithm to obtain the optimal solution for operating the system (iii) evaluating the performance with the optimization solver Gurobi.

The remaining of the paper is organized as follows. Section 2 presents the problem description. We proposes ant colony optimization algorithm (ACO) for the model presented in section 3. Section 4 tests the proposed models in Gurobi Optimizer and the effectiveness of the ACO. Section 5 concludes this research and suggests future research.

2. PROBLEM DESCIPTION

This study considers dock assignment and truck sequence simultaneously at a cross-docking terminal. Since one of the cross-docking benefits is to reduce the inventory cost, the objective of our truck scheduling problem is to minimize the sum of all product stay time costs at the cross-docking terminal. The products' stay time that starts from product's unloading time to its loading time is also minimized. The inventory cost takes into account the quantity and transit time of the product within the terminal. This objective is to maximize the turnover of goods.

2.1 Assumptions

The assumptions in this research are as follows.

1. The products associate of both inbound and outbound vehicles is known a priori. This is especially important for the e-commerce operations.
2. There is more than one inbound and outbound dock. The number of inbound docks is less than the number of inbound vehicles.
3. The vehicle cannot leave the dock until it finishes the loading or unloading operations (no preemption of trucks is allowed).
4. The cross-docking terminal has unlimited temporary storage area to store the products that is not for loading into the outbound vehicle currently at the outbound dock.
5. All inbound and outbound trucks are available at time zero and can be assigned to any inbound dock.
6. The transportation time depends on the relative distance between the inbound dock where the goods are unloaded and the outbound dock where they are loaded.
7. Truck changeover time is the same for all trucks and it is known a priori.
8. The outbound docks are destination-exclusive mode and are positioned at pre-determined shipping dock to ease the internal operation.

2.2 Mathematical Model

Input

C: Set of inbound docks where: $c = 1,2,\ldots,|C|$
g_{kl}: Flow between inbound truck k and outbound truck l
K: Set of inbound trucks, where: $k = 1,2,\ldots,|K|$
L: Set of outbound trucks, where: $l = 1,2,\ldots,|L|$
M: A large number
t_{cd}: Transportation time between inbound docks c and outbound dock d
a: Unit loading or unloading time
λ: Truck changeover time

Decision Variables

s_k: Completion time of inbound truck k.
u_l: Completion time of outbound truck l.
x_{kl}: Completion time of the flow from inbound truck k to outbound truck l.
γ_{cp}: Completion time of inbound truck at position p on dock c.

δ_{cp}: Dock time of inbound truck at position p on dock c.

$$y_{kcp} = \begin{cases} 1 & \text{if inbound truck } k \text{ is assigned to dock } c \text{ in the } pth \text{ position} \\ 0 & \text{otherwise} \end{cases}$$

The mathematical model can be formulated as follows:

$$Min\ Z = \sum_{k \in K} \sum_{l \in L} g_{kl} (u_l - s_k) \tag{1}$$

Subject to

$$\sum_{c \in C} \sum_{p \in K} y_{kcp} = 1 \qquad \forall k \in K \tag{2}$$

$$\sum_{k \in K} y_{kcp} \leq 1 \qquad \forall c \in C, \forall p \in K \tag{3}$$

$$\delta_{c1} = 0 \qquad \forall\, c \in\, C \tag{4}$$

$$\gamma_{cp} = \delta_{cp} + \sum_{k \in K} (y_{kcp} \times \alpha \sum_{l \in L} g_{kl}) \qquad \forall c \in C, \forall p \in K \tag{5}$$

$$\delta_{cp} = \gamma_{c(p-1)} + \lambda \cdot y_{kc(p-1)} \qquad \forall k \in K, \forall c \in C, \forall p \in K, p \neq 1 \tag{6}$$

$$s_k \geq \gamma_{cp} - M(1 - y_{kcp}) \qquad \forall k \in K, \forall c \in C, \forall p \in K \tag{7}$$

$$x_{kl} = \sum_{p \in K} y_{kcp} (\alpha \cdot g_{kl} + t_{cl}) \qquad \forall c \in C, \forall k \in K, \forall l \in L \tag{8}$$

$$u_l \geq Max(x_{kl} + s_k) \qquad \forall k \in K, \forall l \in L \tag{9}$$

$$y_{kcp} \in \{0,1\} \qquad \forall k \in K, \forall c \in C, \forall p \in K \tag{10}$$

The objective function (1) is to minimize the total product inventory cost for products that unloading from inbound trucks to loading to outbound trucks. Constraint (2) guarantees that each inbound truck is assigned to exactly one position at inbound dock c. Constraint (3) enforces that every inbound dock serves at most one truck at a time. Constraint (4) represents the trucks assigned in the first position the starting time is zero. Constraint (5) represents the completion time of inbound trucks as being greater than or equal to its starting time plus its loading time. Constraint (6) represents the starting time of inbound trucks as being greater than or equal to the completion time of the inbound truck assigned to the previous position in the same inbound dock or to the arrival time of the inbound truck assigned to that dock and position. Constraint (7) computes the starting time of each inbound truck. Constraint (8) computes the item transportation time from inbound truck k to outbound truck l. Constraint (9) computes the completion time of each outbound truck.

The truck sequencing problem in cross-docking operations of a single dock has been shown to be NP-hard (Chen and Lee, 2009). The mixed integer programming model takes into account dock assignment and truck sequencing simultaneously. It becomes intractable when the number of inbound and outbound trucks, number of inbound docks, interaction between inbound and outbound trucks increases. For example, when number of inbound and outbound trucks equal to 5, respectively, and the number of inbound docks is 2, the total number of decision variables is 95 and the total number of constraints is 200. Once the number of inbound and outbound trucks becomes 12 and the number of docks equals to 5, the number of decision variables is increased to 1012 and the number of constraints becomes 2436. Exact solution approach cannot solve the problem within reasonable time. We propose the ant colony optimization algorithm to solve the problem and will be described in details in next section.

3. ANT COLONY OPTIMIZATION ALGORITHM

The truck scheduling problem for cross-docking systems is proved as NP-hard (Chen and Song, 2009). The mathematical model developed in previous section is not able to solve the problem even for small size instances. We develop an ant colony optimization algorithm to solve the problem. The ant colony optimization (ACO) was first proposed by Dorigo et al. (1996). Subsequently, many variants of ACO have been developed and applied extensively in the combinatorial optimization problems. Dorigo and Stützle (2004) provided descriptions of available ACO algorithms and related literature review. In principle, ACO can be applied to any discrete optimization problem for which some solution construction mechanism can be conceived.

This section describes the proposed algorithm for solving the truck scheduling problem. The procedures of our ACO are shown in figure 2 and described as follows. Details of each step are introduced in the following subsections.

1 Initialization

Initialize parameters and value of pheromone matrices.

2 Dock assignment process

2.1 Select a vehicle to a dock.
2.2 Select an inbound dock based on state transition rule.
2.3 If all vehicles are assigned, go to step 3; otherwise go to step 2.1.

3 Local updating

3.1 Update the pheromone levels.
3.2 If all ants have solutions, go to step 4; otherwise go to step 2.

4 Local search

Using swap and insertion on the best solution of current iteration. If the iteration best solution is better than the global best solution, update the global best solution.

5 Global updating

Update the pheromone by iteration best solution and best solution till now.

6 Terminating condition

If the terminating criterion (maximum number of iterations in this paper) is met, stop; otherwise repeat steps 2~5.

3.1 Solution Representation

The solution representation separates the inbound and outbound dock assignment. The first row represents inbound vehicles while the second row is for outbound vehicles. The total length of a solution will be represented by $V_I + m_I - 1$ and $V_O + m_O - 1$for inbound and outbound docks, respectively. V_I and V_O are the number of trucks for inbound and outbound, while m_I and m_O represent the number of inbound and outbound docks, respectively. For example, the solution representation illustrated in figure 3 can be decoded as a cross-docking system with 5 vehicles for both inbound and outbound and 2 docks for each direction. For inbound operation, vehicles 1, 4, 3 are assigned to first inbound dock, while vehicles 2 and 5 are assigned to second inbound dock. Similarly, vehicles 5, 3, and 4 are assigned to outbound dock 1, and vehicles 2 and 1 are assigned to second outbound dock. The docking sequence is also determined by such solution representation.

3.2 Solution Construction

In our ACO, we assign a vehicle i to a dock j chosen for ant k by the following state transition rule.

$$s = \begin{cases} arg \max\limits_{j \in N_i} \left\{ (\tau_{ij}) \cdot (\eta_{ij})^{\beta} \right\} & , q \le q_0 \\ S & , q > q_0 \end{cases} \tag{11}$$

$$S: \; P_{ij}^{k}(t) = \begin{cases} \dfrac{(\tau_{ij}) \cdot (\eta_{ij})^{\beta}}{\sum\limits_{q \in N_i} (\tau_{iq}) \cdot (\eta_{iq})^{\beta}} & , if \;\; j \in N_i \\ 0 & , otherwise \end{cases} \tag{12}$$

where N_i is the set of docks which can be used by vehicle i, τ_{ij} is the pheromone of edge (i, j), η_{ij} is defined as the reciprocal of the dock j total use time before assign vehicle i. The idea is assign a vehicle to a less used dock for better dock utilization. β is the parameter that determines the relative effect of τ_{ij} versus η_{ij} ($\beta > 0$), q is a random variable uniformly distributed in [0, 1], and q_0 is a pre-defined parameter ($0 \le q_0 \le 1$). If $q \le q_0$, then the best node j for customer i is determined according to eq. (11). On the contrary, it is chosen according to S which is a random variable selected according to the probability distribution given in eq. (12). Hence, the parameter q_0 determines the relative importance of exploitation eq. (11) versus exploration eq. (12).

3.3 Pheromone Update

The pheromone updating of a typical ACO includes global and local updating rules. The ants apply a local pheromone update rule immediately after they crossed an edge (i, j) during the tour construction. The local pheromone updating rule of our ACO is

$$\tau_{ij}^{new} = (1-\rho)\cdot\tau_{ij}^{old} + \rho\cdot\tau_0 \text{, if } \{\text{edge}(i,j)\in T_h\} \tag{13}$$

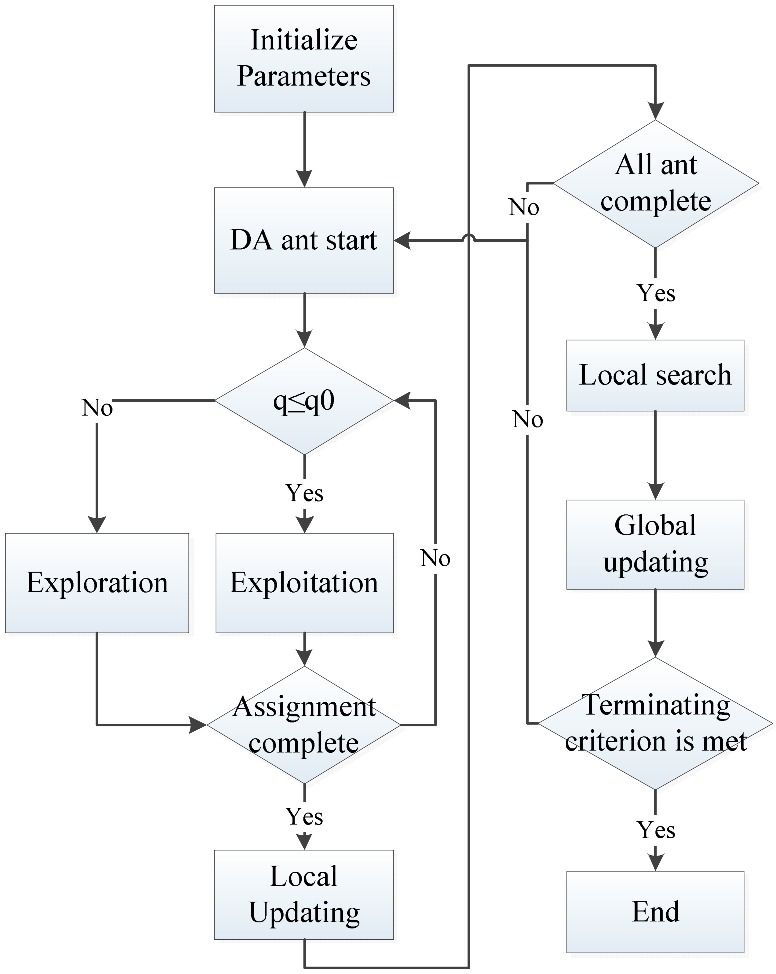

Figure 2: The flowchart of the proposed ACO

1	4	3	0	2	5
5	3	4	0	2	1

Figure 3: A representation of solution

where T_h denotes the assignment solution constructed by ant h, ρ is the pheromone decay parameter in the range of [0, 1] that regulates the reduction of pheromone on the edges. The τ_0 is the initial value of the pheromone matrix for the route construction rule, and is set to be 0.2 in this paper.

In our ACO, the best elitist assignments, including the global-best assignment (T_b) and the iteration-best assignment (T_s), are allowed to lay pheromone on the edges that belong to them. The idea here is to balance between exploitation (through emphasizing the global-best tour) as well as exploration (through the emphasis to the iteration-best tour). The global updating rule of ACO is described as follow.

$$\tau_{ij}^{new} = (1-\rho)\cdot\tau_{ij}^{old} + \rho\cdot\Delta\tau_{ij} \tag{14}$$

where

$$\Delta\tau_{ij} = \begin{cases} 1/L_b & \text{if } (i,j)\in T_b \\ 1/L_s & \text{if } (i,j)\in T_s \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

L_b and L_s denote the objective function value of the global-best solution and the iteration-best solution of the problem, respectively. T_b and T_s are the global best solution and iteration-best solution, respectively.

3.4 Local Search

Local search is a time-consuming procedure of ACO. The analyses in Ting and Chen (2013) showed that it is efficient for ACO to only apply local search to the best solution among all solutions built at the current iteration. To save the computation time, only the iteration-best solution is applied local search in this paper. In addition, two local search methods are involved in our ACO, including swap and insertion. The local search could be applied within a dock or between docks. This is because that diverse neighborhood moves can expand the solution searching space. Two vehicles are exchanged in swap. Insertion is to move one vehicle from its current position to another position, in the same route or in a different route. In each iteration, we randomly implement only one local search method. Thus, we assume that every approach has the same probability to be selected for local search.

4. COMPUTATIONAL EXPERIMENTS

The proposed ant colony optimization algorithm was coded in Microsoft Visual Studio C++ 2010 and run on a PC with an i5-2400 3.10GHz processor, 8.0GB of RAM and Windows 7 operating system. The instances were randomly generated based on the setting of Liao *et al.* (2013) for the dock assignment problem with cross-docking. The number of inbound docks ranging between 2 and 5 is smaller than the number of inbound vehicles, while the number of outbound docks is equal to the number of outbound vehicles. Each test instance is replicated 10 times and tested using different random seeds in each run. The proposed ACO was tested and compared with Gurobi optimizer 6.05 with the running time of two hours.

In preliminary experiments we tried to find a good parameter setting for the proposed ACO algorithm. We consider a set of parameters for the algorithm and then modifying one at a time, while keeping the others fixed. The parameters that were tested include: $\beta \in \{2, 3, 4\}$, $\rho \in \{0.1, 0.2, 0.3\}$, $q_0 \in \{0.1, 0.2, 0.3\}$, $b \in \{30, 40, 50\}$, $Iter \in \{100, 300, 500\}$. We found that for the parameter setting, $\beta = 2$, $\rho = 0.2$, $q_0 = 0.5$, $b = 50$, $Iter = 500$ can provide the best average solution. These parameters will be used for all instances for further experiment.

Table 1 shows the results of our proposed ACO and Gurobi solution. Column 1 is the instance number. Columns 2~4 are the number of docks and number of vehicles used for inbound and outbound docks by this instance. For example, the first instance has two inbound docks and five inbound trucks, and five outbound docks with five outbound trucks. The Gurobi solution is either the optimal solution marked by '*' or the best solution found during 7200 seconds limit. For the proposed ACO, each instance is presented with best solution, and average run times in seconds. Among 24 instances, Gurobi can obtain optimal solutions in 15 smaller size instances which our ACO can also provide the same solution. For those 9 larger size instances that Gurobi cannot provide the optimal solution, our ACO can either obtain the same solution or provide better result. Our ACO can improve the solutions obtained by Gurobi by 0.32% on average, while the computational time is much smaller than that of Gurobi.

Table 1: Computation results of the proposed ACO and Gurobi

Ins.	Inbound dock	Outbound dock	Truck	Gurobi		ACO		GAP(%)
				Opt.	Time	Best	Time	
DA-1	2	5	5	510*	1.37	510	0.12	0.00
DA-2	2	6	6	840*	1.56	840	0.17	0.00
DA-3	2	7	7	767*	22.8	767	0.22	0.00
DA-4	2	8	8	2446*	60.3	2446	0.28	0.00
DA-5	2	10	10	3730	7200	3730	0.41	0.00
DA-6	2	12	12	12001	7200	11973	0.57	-0.23
DA-7	3	5	5	404*	3.0	404	0.14	0.00
DA-8	3	6	6	663*	3.9	663	0.19	0.00
DA-9	3	7	7	646*	22.4	646	0.24	0.00
DA-10	3	8	8	2002	7200	2002	0.31	0.00
DA-11	3	10	10	2906	7200	2877	0.45	-1.00
DA-12	3	12	12	8817	7200	8531	0.63	-3.24
DA-13	4	5	5	385*	2.1	385	0.16	0.00
DA-14	4	6	6	651*	2.2	651	0.22	0.00
DA-15	4	7	7	574*	92.6	574	0.28	0.00
DA-16	4	8	8	1489*	205.0	1489	0.34	0.00
DA-17	4	10	10	2447	7200	2414	0.49	-1.35
DA-18	4	12	12	6731	7200	6694	0.68	-0.55
DA-19	5	5	5	304*	0.7	304	0.18	0.00
DA-20	5	6	6	539*	12.8	539	0.25	0.00
DA-21	5	7	7	561*	29.9	561	0.31	0.00
DA-22	5	8	8	1566*	144.2	1566	0.38	0.00
DA-23	5	10	10	2027	7200	2004	0.54	-1.13
DA-24	5	12	12	6603	7200	6590	0.73	-0.20
Average				2483.7	2725.2	2465.0	0.35	-0.32

*optimal solution is found.

4.1 The Impact of Local Search

The traditional ACO does not have local search. To show the impact of implementing the local search, we run ACO without local search for the 24 instances. The results of the comparison are shown in Table 2. In table 2, column 1 is the test instance number, columns 2-5 are the best results and CPU times in seconds for ACO without and ACO with local search, respectively. Column 6 is the improvement percentage for implementing local search in the ACO. The ACO without local search can obtain those optimal solutions that Gurobi found within two hours limitation. There are five instances that ACO without local search found worse solutions than those obtained by Gurobi. This might indicate that the ACO without local search could be trapped at local optimum. However, ACO with local search can improve six instances up to 2.35% with 0.35% improvement on average. In the larger instances, ACO with local search can provide better solutions with a slightly longer computational time.

4.2 Comparison with Makespan Objective

To further investigate the objective function that we discussed in this paper, we modify the objective function to minimize makespan as used in previous research (Chen and Lee, 2009). The proposed ACO was implemented to solve the problem as well by changing the objective function. We compare the resulting inventory holding cost and makespan for the objective in minimizing the makespan with the original objective of minimizing the inventory holding cost.

In table 3, column 1 is the instance number, columns 2-5 are the inventory holding cost and makespan under the objective of minimizing inventory cost and minimizing makespan, respectively. Columns 6-7 are the difference for the inventory holding cost and makespan for different objective functions as computed in eq. (16).

$$Diff = \frac{(Obj_I - Obj_M)}{Obj_M} \times 100 \tag{16}$$

where the objectives are inventory cost and makespan, respectively. There are two instances that different objectives provide same inventory cost and makespan. As we expected, lower inventory cost with larger makespan are obtained for minimizing inventory objective. The difference in inventory cost ranges from 0.00 to -17.65% with average at -7.77%. However, the makespan difference ranges from 0.00 to 13.33% with average at 4.18%. From the supply chain perspective, the objective of minimizing inventory holding cost might provide better results than that of minimizing makespan.

Table 2 results comparison for ACO with and without local search

Ins.	BKS	ACO w/o LS		ACO w/ LS		Imp (%)
		Best.	Time	Best	Time	
DA-1	510*	510	0.09	510	0.12	0.00
DA-2	840*	840	0.12	840	0.17	0.00
DA-3	767*	767	0.15	767	0.22	0.00
DA-4	2446*	2446	0.19	2446	0.28	0.00
DA-5	3730	3730	0.28	3730	0.41	0.00
DA-6	12001	12118	0.40	11973	0.57	-1.20
DA-7	404*	404	0.10	404	0.14	0.00
DA-8	663*	663	0.13	663	0.19	0.00
DA-9	646*	646	0.17	646	0.24	0.00
DA-10	2002	2002	0.21	2002	0.31	0.00
DA-11	2906	2877	0.32	2877	0.45	0.00
DA-12	8817	8565	0.44	8531	0.63	-0.40
DA-13	385*	385	0.11	385	0.16	0.00
DA-14	651*	651	0.15	651	0.22	0.00
DA-15	574*	574	0.20	574	0.28	0.00
DA-16	1489*	1489	0.24	1489	0.34	0.00
DA-17	2447	2472	0.35	2414	0.49	-2.35
DA-18	6731	6757	0.48	6694	0.68	-0.93
DA-19	304*	304	0.13	304	0.18	0.00
DA-20	539*	539	0.17	539	0.25	0.00
DA-21	561*	561	0.22	561	0.31	0.00
DA-22	1566*	1566	0.26	1566	0.38	0.00
DA-23	2027	2038	0.38	2004	0.54	-1.67
DA-24	6603	6667	0.51	6590	0.73	-1.15
Average	2483.7	2482.1	0.24	2465.0	0.35	-0.35

5. CONCLUSION

The truck scheduling problem involved in the cross-docking system is one of the important decisions for the cross-docking system efficiency. Good dock assignment can reduce the transit time between inbound and outbound docks and further reduce the dock delays and inventory cost. In this paper, we integrated the truck scheduling and dock assignment of inbound trucks for the cross-docking operations. The objective is to reduce the total time for each product within the cross-docking terminal. A mixed integer programming model is formulated. Since this integrated problem is NP-hard, we developed an ant colony optimization algorithm to solve the problem. To test the proposed ACO algorithm, 24 test instances were randomly generated based on the setting from the literature. The instances were also solved by the Gurobi optimization solver with time limit of 2 hours. The computational results show that our ACO can obtain the optimal solutions for small size instances and improve the solutions obtained by Gurobi with much shorter computational times.

We also compared the resulting inventory cost and makespan with the objective of minimizing makespan. By minimizing the inventory holding cost as the objective, the makespan will increased by 4.18%, while the inventory holding cost can reduce by 7.77%. This indicates that minimizing inventory holding cost could be an objective for the supply chain. Our ACO is also an effective and suitable approach to tackle the multi-dock assignment problem. In the future, we could work on the multi-dock problem with time window constraints. Another extension might to consider multiple-objective truck scheduling problem for the multiple dock cross-docking terminal. Other possible extension would to integrate the vehicle routing problem with dock assignment problem such that both cross-docking operational issues could be solved simultaneously.

Table 3 Comparison for inventory cost and makespan based on different objective functions.

Obj. / Ins.	Inventory		Makespan		Diff. (%)	
	Inv.	MS	Inv.	MS	Inv.	MS
DA-1	510	46	602	44	-15.28	4.55
DA-2	840	51	840	51	0.00	0.00
DA-3	767	59	870	54	-11.84	9.26
DA-4	2446	82	2719	79	-9.30	3.80
DA-5	3730	102	4311	101	-13.48	0.99
DA-6	11973	160	13234	158	-9.53	1.27
DA-7	404	32	471	31	-14.23	3.23
DA-8	663	36	699	35	-5.15	2.86
DA-9	646	42	708	39	-8.76	7.69
DA-10	2002	60	2120	57	-5.57	5.26
DA-11	2877	70	3233	68	-11.01	2.94
DA-12	8531	108	9395	106	-9.20	1.89
DA-13	385	30	441	29	-12.70	3.45
DA-14	651	34	659	34	-1.21	0.00
DA-15	574	34	697	30	-17.65	13.33
DA-16	1489	43	1617	42	-7.92	2.38
DA-17	2414	60	2673	57	-9.69	5.26
DA-18	6694	84	7006	82	-4.45	2.44
DA-19	304	19	305	18	-0.33	5.56
DA-20	539	30	539	30	0.00	0.00
DA-21	561	31	592	28	-5.24	10.71
DA-22	1566	43	1637	40	-4.34	7.50
DA-23	2004	46	2035	44	-1.52	4.55
DA-24	6590	78	7167	77	-8.05	1.30
Average	2465.00	57.5	2690.42	55.6	-7.77	4.18

REFERENCES

1. Saddle Creek Corp. (2011). 2011 Cross-Docking Trends Report. Saddle Creek Corporation Whitepaper series..
2. Agustina, D., Lee, C.K.M. and Piplani, R. (2010). A Review: Mathematical Models for Cross Docking Planning. *International Journal of Engineering Business Management*, 2: 47-54.
3. Alpan, G., Ladier, A.L., Larbi, R. and Penz, B. (2011). Heuristic Solutions for Transshipment Problems in a Multiple Door Cross Docking Warehouse. *Computers & Industrial Engineering*, 61: 402-408.
4. Apte, U.M. and Viswanathan, S. (2000). Effective Cross Docking for Improving Distribution Efficiencies. *International Journal of Logistics Research and Applications*, 3: 291-302.
5. Boloori Arabani, A., Fatemi Ghomi S.M.T. and Zandieh, M. (2011). Meta-heuristics Implementation for Scheduling of Trucks in a Cross-docking System with Temporary Storage. *Expert Systems with Applications*, 38: 1964-1979.
6. Boysen, N. and Fliedner, M. (2010). Cross Dock Sequencing: Classification, Literature Review and Research Agenda. *Omega*, 38: 413-422.
7. Bozer, Y.A. and Carlo, H.J. (2008). Optimizing Inbound and Outbound Door Assignments in Less-than-truckload Crossdocks, *IIE Transactions*, 40: 1007-1018.
8. Buijs, P., Vis, I.F.A. and Carlo, H.J. (2014). Synchronization in Cross-docking Networks: A Research Classification and Framework. *European Journal of Operational Research*, 239: 593-608.
9. Chen, F. and Song, K. (2009). Minimizing Makespan in Two-stage Hybrid Cross Docking Scheduling Problem. *Computers & Operations Research*, 36: 2066-2073.
10. Chen, F. and Lee, C. Y. (2009). Minimizing the Makespan in a Two-machine Cross-docking Flow Shop Problem. *European Journal of Operational Research*, 193: 59-72.
11. Cohen, Y. and Keren, B. (2009). Trailer to Door Assignment in a Synchronous Cross-dock Operation.

International Journal of Logistics Systems and Management, 5: 574-590.

12. Dorigo, M., Maniezzo, V. and Colorni, A. (1996). Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 26: 29-41.
13. Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization. Bradford Books*, MIT Press, Cambridge, MA.
14. Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, NY.
15. Ladier, A.-L. and Alpan, G. (2016). Cross-docking Operations: Current Research versus Industry Practice, *Omega*, 62: 145-162.
16. Li, Y., Lim, A. and Rodrigues, B. (2004). Crossdocking-JIT Scheduling with Time Windows. *Journal of the Operational Research Society*, 55: 1342-1351.
17. Liao, T.W., Egbelu, P.J. and Chang, P. C. (2012). Two Hybrid Differential Evolution Algorithms for Optimal Inbound and Outbound Truck Sequencing in Cross Docking Operations. *Applied Soft Computing*, 12: 3683-3697.
18. Liao, T.W., Egbelu, P.J. and Chang, P. C. (2013). Simultaneous Dock Assignment and Sequencing of Inbound Trucks under a Fixed Outbound Truck Schedule in Multi-door Cross Docking Operations. *International Journal of Production Economics*, 141: 212-229.
19. Miao, Z., Lim, A. and Ma, H. (2009). Truck Dock Assignment Problem with Operational Time Constraint within Crossdocks. *European Journal of Operational Research*, 192: 105-115.
20. Miao, Z., Cai, S. and Xu, D. (2014). Applying an Adaptive Tabu Search Algorithm to Optimize Truck-dock Assignment in the Crossdock Management System. *Expert Systems with Applications*, 41: 16-22.
21. Material Handling Institute, Glossary http://www.mhi.org/glossary, access date: 2017/4/15.
22. Napolitano, M. (2000). *Making the Move to Cross Docking: A Practical Guide to Planning, Design, and Implementing a Cross Dock Operation*. Warehousing Education and Research Council, Oak Brook, Illinois.
23. Oh, Y., Hwang, H., Cha, C. N. and Lee, S. (2006). A Dock-door Assignment Problem for the Korean Mail Distribution Center. *Computers & Industrial Engineering*, 51: 288-96.
24. Shuib, A. and Fatthi, WNAWA. (2012). A Review on Quantitative Approaches for Dock Door Assignment in Cross-docking. *International Journal on Advanced Science, Engineering and Information Technology*, 2: 30-34.
25. Stalk, G., Evans, P. and Shulman, L.E. (1992). Competing on Capability: The New Rules of Corporate Strategy. *Harvard Business Review*, 70: 57-69.
26. Stephan, K. and Boysen, N. (2011). Cross-docking. *Journal of Management Control*, 22: 129-137.
27. Tsui, L.Y. and Chang, C.H. (1990), A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards. *Computers & Industrial Engineering*, 19: 309-312.
28. Tsui, L.Y. and Chang, C.H. (1992). An Optimal Solution to a Dock Door Assignment Problem. *Computers & Industrial Engineering*, 23: 283-286.
29. Walha, F., Chaabane, S., Bekrar, A. and Loukil, T. (2014). The Cross Docking under Uncertainty: State of the Art. *2014 International Conference on Advanced Logistics and Transport* (*ICALT*), Hammamet, Tunisia, 330-335.
30. Witt, C.E. (1998). Crossdocking: Concepts Demand Choice. *Material Handling Engineering*, 53: 44-49.
31. Yu, W. (2015). Truck Scheduling for Cross Docking Systems with Multiple Receiving and Shipping Docks. *International Journal of Shipping and Transport Logistics*, 7: 174-196.
32. Yu, W. and Egbelu, P.J. (2008). Scheduling of Inbound and Outbound Trucks in Cross Docking Systems with Temporary Storage. *European Journal of Operational Research*, 184: 377-396.