Computers & Industrial Engineering 59 (2010) 85–92

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Ant colony optimization algorithm to solve for the transportation problem of cross-docking network

Rami Musa[a,*], Jean-Paul Arnaout[b], Hosang Jung[c]

[a] Supply Chain Solutions Group Agility Logistics, 1995 North Park Place, Suite: 310 Atlanta, GA 30339, United States
[b] Industrial and Mechanical Engineering Department at the Lebanese American University, Byblos, Lebanon
[c] Department of Management Engineering, Sangmyung University, Anseodong 300, Dongnamku, Cheonan, Choongnam 330-720, South Korea

ARTICLE INFO

Article history:
Received 21 April 2009
Received in revised form 7 March 2010
Accepted 9 March 2010
Available online 12 March 2010

Keywords:
Cross-docking
Transportation planning
Ant colony optimization (ACO)
Metaheuristics
Integer programming

ABSTRACT

This paper addresses the transportation problem of cross-docking network where the loads are transferred from origins (suppliers) to destinations (retailers) through cross-docking facilities, without storing them in a distribution center (DC). We work on minimizing the transportation cost in a network by loading trucks in the supplier locations and then route them either directly to the customers or indirectly to cross-docking facilities so the loads can be consolidated. For generating a truck operating plan in this type of distribution network, the problem was formulated using an integer programming (IP) model and solved using a novel ant colony optimization (ACO) algorithm. We solved several numerical examples for verification and demonstrative purposes and found that our proposed approach finds solutions that significantly reduce the shipping cost in the network of cross-docks and considerably outperform Branch-and-Bound algorithm especially for large problems.

1. Introduction

Efficient control of the physical flow of the supply chain has been regarded as one of the most important aspects for minimizing the total cost of supply chains. Since 30% of an item price is incurred in the distribution process, lots of companies are currently trying to develop new distribution strategies to efficiently manage their product flow (Apte & Viswanathan, 2000). Among many strategies developed so far, cross-docking is believed to be an efficient strategy to minimize unnecessary inventory and to reduce service cycle times. Cross-docking attempts to lessen or even eliminate such burdens by reducing warehouses to purely trans-shipment centers where receiving and shipping are its only functions (Li, Lim, & Rodrigues, 2004). In other words, in the cross-docking network, the warehouses, as cross-docks, are transformed from inventory repositories to points of delivery, consolidation and pick-up (Chen, Guo, Lim, & Rodrigues, 2006). As material (i.e. products) is unloaded from one vehicle and immediately reloaded onto another vehicle, the models for generating a distribution plan of the cross-docking network must explicitly account for individual transportation units (Donaldson, Johnson, Ratliff, & Zhang, 1999).

Until now, lots of studies regarding cross-docking networks have been reported in the literature, and they can be divided into two different categories: (1) dock-door assignment problems and (2) distribution planning problems of a cross-docking network. In the dock-door assignment problem, the focus lies on how to find a proper way of assigning each product received from incoming doors (receiving doors) to outgoing doors (shipping doors) in order to minimize the total material handling cost of the products. Tsui and Chang (1992) presented a general model of the dock-doors assignment, and then developed a solution based on the Branch-and-Bound (B&B) algorithm. Also, the staging of products in a cross-dock to avoid floor congestion and increase throughput has also been studied (Bartholdi, Gue, & Kang, 2001; Gue & Kang, 2001). As for the distribution planning problems of a cross-docking network, Donaldson et al. (1999) studied a schedule-driven transportation planning in the cross-docking network and Ratliff, Vate, and Zhang (1999) studied the automobile delivery network to generate a distribution plan by determining the ideal number and location of cross-docks together. In addition, Li et al. (2004) studied a cross-dock operation which aims to eliminate or minimize storage and order picking activity in the cross-dock using just-in-time (JIT) scheduling. They convert their problem to a machine scheduling problem that allows a warehouse to function as a cross-dock where transit storage time for cargo is eliminated or minimized. More recently, Chen et al. (2006) studied distribution planning for the cross-docking network taking into consideration delivery and pick-up time windows, warehouse capacities and inventory-handling costs. They also proposed both simulated annealing and Tabu search heuristics as solution methods because of the

* Corresponding author.
E-mail addresses: rmusa@agilitylogistics.com, rami.musa@gmail.com (R. Musa), jparnaout@lau.edu.lb (J.-P. Arnaout), hsjung@smu.ac.kr (H. Jung).

0360-8352/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.03.002

complexity of their problem. The details on various heuristics for optimization can be found in Pardalos and Resende (2002) and Pardalos and Romeijn (2002).

In this paper, we study a distribution planning problem of the cross-docking network to determine the best way to load and route the trucks in the network for minimizing the final shipping cost. The problem we handle in this research is adopted from that of Donaldson et al. (1999). The integer programming model representing the problem is also adopted but modified for our consideration, which can determine how many trucks should be assigned to each link (origin to destination, origin to cross-dock, and cross-dock to destination) and how flow should be routed. Basically, this transportation problem of cross-docking network in this research can be classified as a pure-integer programming model containing both non-negative integer variables and binary integer variables. It is well known that a binary integer programming model is NP-hard. Thus, because every integer programming model can be written as a binary integer programming model by transforming one to the other in polynomial time (Garey & Johnson, 1979; Papadimitriou & Steiglitz, 1998), any integer programming model including our problem in this research is also NP-hard. In this research, we developed an ant colony optimization based heuristics for solving this NP-hard problem.

Ant colony optimization is a metaheuristic that was inspired by the social behavior of ants in finding the shortest paths from their nest (colony) to a food source (Dorigo, Maniezzo, & Colorni, 2006). Although ants cannot see the *big picture* when they move over different paths, they end up following the shortest path by coordinating with each other through *depositing* and *sniffing* a chemical substance called *pheromone.* The deposited pheromone evaporates with time. Therefore, after trying few tours from and to the nest, more pheromone will be condensed in the short paths compared to the longer paths. This is because the deposited pheromone evaporates faster in the longer path. Consequently, more ants will be attracted to the shorter paths as time progresses. ACO has been successfully used to solve combinatorial problems such as the *Traveling Salesman Problem*, *Quadratic Assembly Problem* and *Vehicle Routing Problem* (Dorigo & Stützle, 2004). To solve an optimization problem using ACO, the problem has to be represented by a *connected graph* (nodes and edges). Initially, we deposit a certain amount of pheromone in each edge in the graph. After that, ants move from one node to another according to a rule that determines the preference for an ant to move from its current node to another one. The probability to choose a next node is determined by two factors: pheromone amount (τ) and visibility (η).

After all the ants construct their tours, pheromone will be updated locally and globally according to the quality of the constructed tour solutions (cost value) and the evaporation rate (ρ). There are different parameters that govern the performance of an ACO algorithm such as: number of ants, greedy heuristic to determine the visibility, evaporation rate, and the importance of pheromone versus visibility.

The structure of this article is organized in five sections. In Section 2, we introduce the problem statement and state the assumptions and the mathematical model of the transportation problem of the cross-docking network. Our ACO-heuristic approach is presented in details in Section 3. Next, we solve several numerical examples in Section 4 using our proposed approach to show the superiority of its solutions. In Section 5, we summarize our work and explain some of our future research ideas where some assumptions can be easily relaxed.

2. Problem statement

For a network made of I suppliers, J customers, and K cross-dock facilities (see Fig. 1); our objective is to find the best fleet dispatching and consolidation plans so the total transportation cost is minimized. We do so by determining the loads to be sent directly from the suppliers to the customers and the amounts to be sent indirectly through the cross-docking facilities. The loads sent to the cross-dock facilities are consolidated and sent accordingly to their final destination.

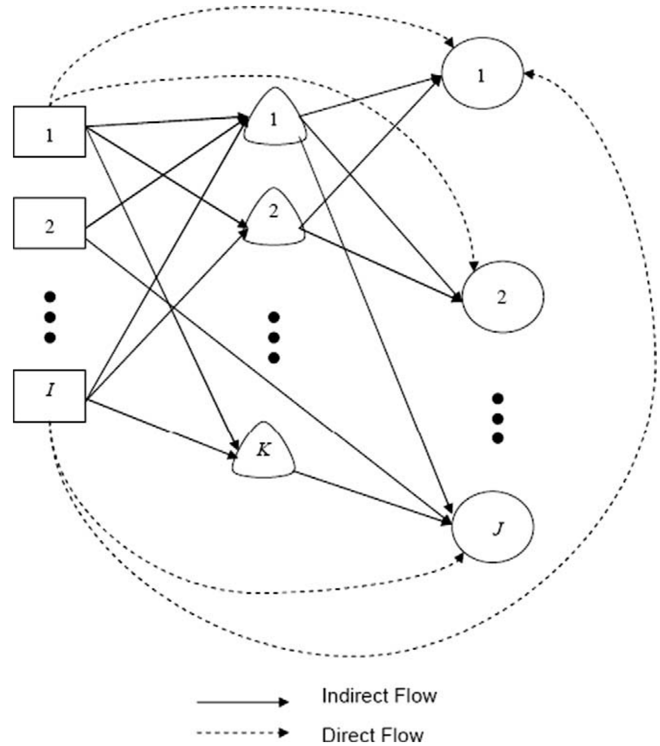

Fig. 1. Transportation problem of the cross-docking network.

The mathematical model we have adopted and enhanced assumes the following:

1. The loads to be sent from a supplier location i to a final customer j (S_{ij}) are known and assumed to be less than the truck capacity (C); i.e. $S_{ij} \leqslant C \quad \forall i = 1, \ldots, I, \forall j = 1, \ldots, J$. Otherwise, the solution is trivial since the truck for that flow will need to go directly from i to j.
2. Trucks are always available when needed. Now, most of manufacturers maintain the transportation contract with independent third-party-logistics (3PL) providers. Under this contract, the number of trucks given by the 3PL provider is determined by considering the maximum possible request of the manufacturer. Thus, the transportation capacity is large enough for the manufacturer in this case. Of course, when facing any unexpected shortage regarding the transportation availability of the 3PL provider, the manufacturer can search and find alternative providers with a little higher cost. In short, the availability of trucks is not a fixed constraint, and we assume abundance of trucks in this research. In addition, the authors are now undertaking the successive research works on multiple-period transportation planning, and some issues on the restricted availability of transportation will be discussed in depth in those research works.
3. All the trucks have the same capacity (C). This goes with the previous assumption. If the previous assumption is relaxed, then we could simply add an index to each truck for their capacities.
4. The shipped loads are assumed to have the same cross sections that can utilize the whole frontal cross section of the truck. The shipped quantities are expressed by the length they need to

utilize the overall length of the truck. This reduces the complexity of the shipped items from their three dimensional shapes into single dimensional shapes.

5. The shipping costs are associated with both the traveled distances and traveling time among locations (i.e. origins, cross-docks, and customers). Even if the traveling time is directly related to the traveled distance in general, some routes might have longer or shorter traveling time due to the road characteristics (e.g. bottleneck, expressway) or the road condition (e.g. a good or bad maintenance). Thus, we basically set the shipping cost to be proportional to the traveled distance, but the shipping cost for some special routes has been set considering the traveling time instead of the distance.

As mentioned in the previous section, the mathematical model is adopted from Donaldson et al. (1999). It was modified to fit our scope. Before proceeding to the model, necessary parameters and variables are defined as follows:

I: the set of origin nodes,
J: the set of destination nodes, and
K: the set of cross-docking centers.

$$z_{ij}^k = \begin{cases} 1, & \text{if load from}\, i\, \text{to}\, j\, \text{is sent through cross-docking facility}\, k \\ 0, & \text{otherwise} \end{cases}$$

O_{ik}: a non-negative integer variable representing the number of trucks on link ik, from origin i to cross-dock k
D_{kj}: a non-negative integer variable representing the number of trucks on link kj, from cross-dock k to destination j
R_{ij}: a binary integer variable representing whether to assign the truck on link ij, from origin i to destination j
c_{ij}: the cost of a truck from location i to location j
C: the truck capacity
S_{ij}: the flow at origin i for destination j

Using the above notations, the mathematical model can be formulated as follows:

$$\min \quad \sum_i \sum_j R_{ij} c_{ij} + \sum_i \sum_k O_{ik} c_{ik} + \sum_k \sum_j D_{kj} c_{kj} \tag{1}$$

$$\text{s.t.}: \quad R_{ij} + \sum_k z_{ij}^k = \min\{1, S_{ij}\} \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{2}$$

$$\sum_j S_{ij}.z_{ij}^k \le O_{ik}.C \quad \forall i \in \{1,\ldots,I\},\ \ k \in \{1,\ldots,K\} \tag{3}$$

$$\sum_i S_{ij}.z_{ij}^k \le D_{kj}.C \quad \forall k \in \{1,\ldots,K\},\ \ j \in \{1,\ldots,J\} \tag{4}$$

$$R_{ij}, z_{ij}^k \in \{0,1\} \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\},\ \ k \in \{1,\ldots,K\} \tag{5}$$

The objective function (1) is modeled to minimize the total transportation cost that is made of the direct (from i to j) and indirect transportation costs (from i to k and k to j). Constraints (2) ensure that all the destinations need to be satisfied either through the direct link or through one of the cross-docking centers. If there is no flow from i to j, then there will be no allocation for direct or indirect trips. Constraints (3) and (4) show that all loads are moved via the truck which has a fixed capacity. Constraints (2) are further decomposed using the following constraints (6)–(12) for the ease of running the program using commercial optimization software. We introduced additional binary decision variables (y_{ij}) and a suitably large number (M).

$$R_{ij} + \sum_k z_{ij}^k \le 1 \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{6}$$

$$R_{ij} + \sum_k z_{ij}^k \le S_{ij} \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{7}$$

$$R_{ij} + \sum_k z_{ij}^k \ge 1 - M \cdot y_{ij} \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{8}$$

$$R_{ij} + \sum_k z_{ij}^k \le 1 + M \cdot y_{ij} \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{9}$$

$$R_{ij} + \sum_k z_{ij}^k \ge S_{ij} - M \cdot (1 - y_{ij}) \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{10}$$

$$R_{ij} + \sum_k z_{ij}^k \le S_{ij} + M \cdot (1 - y_{ij}) \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{11}$$

$$y_{ij} \in \{0,1\} \quad \forall i \in \{1,\ldots,I\},\ \ j \in \{1,\ldots,J\} \tag{12}$$

3. Approach

Let us define the trail of the ants by the following pheromone: $\tau_{i,j,k}$ that indicates the favorability of sending demanded load from supplier location i to customer j through cross-dock location k, where $i = 1, \ldots, I$, $j = 1, \ldots, J$, $k = 1 \ldots, K+1$ ($K+1$ indicates that the load to be sent *directly* without passing through any cross-docking facility).

Our ant colony algorithm is explained in the flowchart depicted in Fig. 2. We start the search by placing a certain amount of pheromone in all the $I \cdot J + J \cdot K + I \cdot K$ links. The probability of sending the loads from origin i to final destination j through cross-dock k by ant is given by:

$$P_{i,j,k} = \frac{[\tau_{i,j,k}]^\alpha . [\eta_{i,j,k}]^\beta}{\sum_{g=1}^{K+1} [\tau_{i,j,g}]^\alpha \cdot [\eta_{i,j,g}]^\beta} \tag{13}$$

where,

α, β: exponents that determine the relative effect of the pheromone and greedy amounts on the probability estimate;
$\tau_{i,j,k}$: pheromone that indicates the favorability of sending the shipment from origin i to final destination j through cross-dock k.
$\eta_{i,j,k}$: greedy-heuristic amount that indicates the favorability of sending the shipment from origin i to final destination j through cross-dock k. We define it as the reciprocal of the cost of sending a shipment. If the option is to send the item directly ($k = K + 1$), then $\eta_{i,j,k} = \frac{1}{c_{ij}}$. On the other hand, if the option is to send it through a cross-dock distribution center, then $\eta_{i,j,k} = \frac{1}{c_{ik}/n + c_{kj}/m}$, where m and n are the number of items that can be fit in an empty truck to and from the cross-dock k; respectively.

For each ant, we use the probability function shown in Eq. (13) to generate an $I \times J$ matrix: B. This matrix is to be populated by the cross-dock numbers ($k = 1, \ldots, K + 1$). Remember that $K + 1$ indicates sending the load *directly* without passing through the cross-dock facilities. For example; suppose we have three sources ($I = 3$), five destinations ($J = 5$), and two possible cross-docking facilities ($K = 2$). To populate the cell in the first column and first row in the decision matrix B, we estimate the probabilities for the load from $i = 1$ to $j = 1$ to be transferred through $k = 1$, 2, or 3 ($K + 1$) using the probability function (13). Next, we generate the cumulative density function (CDF) out of the found three probability values. The most probable value is then selected by generating a uniformly distributed random number between 0 and 1 that will be mapped in the y-axis of the CDF to find the associated value in the x-axis ($k = 1$, 2, or 3 ($K + 1$)). Suppose the found probabilities for the three choices ($k = 1$, $k = 2$, and $k = 3$) are respectively: 0.1, 0.4, and 0.6. This means it is more probably

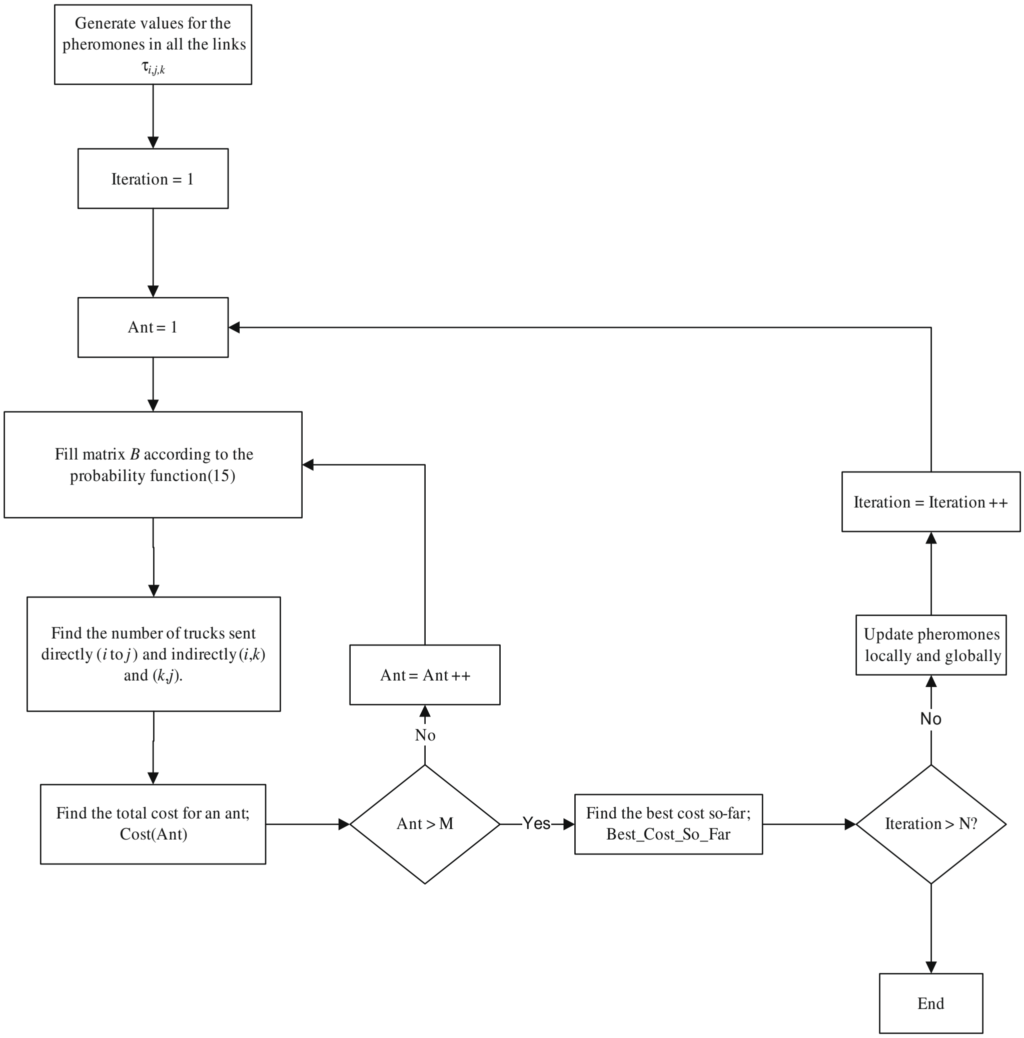

Fig. 2. Algorithm flowchart.

to select the direct route ($k = 3$). The CDF values will be: 0, 0.1, 0.5, and 1. If generate a uniformly distributed random number between 0 and 1 that was found to be 0.7, then we will select the direct route ($k = 3$) because that generated (0.7) value lies in the last range (between 0.5 and 1). Notice that if we repeat that experiment 100 times, then we will end up choosing the three choices around 10, 30, and 60 times. We repeat the same procedure to populate the whole B matrix. An illustration of a B matrix for this example is shown below. The matrix indicates that loads from $i = 1$ to $j = 2$, $i = 2$ to $j = 1$, $i = 2$ to $j = 2$, $i = 3$ to $j = 2$, $i = 3$ to $j = 4$, and $i = 3$ to $j = 5$ are sent directly from origins to destinations since the values associated with those flows are equal to $K + 1$. However, the other loads are sent to the cross-docking locations to be consolidated there and then finally sent to their final destinations. After that, we determine the number of trucks to be sent directly from i to j and indirectly from i to k and k to j. The number of trucks in all the flows can be estimated from the matrix. If the value in the cell associated with i and j is equal to $K + 1$ (direct flow), then we simply send one truck only. The number of the trucks sent indirectly can be estimated from the rows and columns. For instance, we can learn from the first row that the number of trucks sent from $i = 1$ to the cross-docking facility $k = 2$ is equivalent to $\frac{S_{11}+S_{13}+S_{14}+S_{15}}{C}$. On the other hand, we can learn from the second column that the number of trucks sent from $k = 3$ to $j = 2$ is equivalent to $\frac{S_{12}+S_{22}+S_{32}}{C}$.

$$B = \begin{bmatrix} 2 & 3 & 2 & 2 & 2 \\ 3 & 3 & 1 & 1 & 1 \\ 2 & 3 & 2 & 3 & 3 \end{bmatrix}$$

Next, the cost function is found accordingly for each ant. When all the M ants finish generating their B matrices, the pheromone amounts are updated locally and globally according to the quality of the solution; see the following Eqs. (14) and (15). This procedure iterates until a stopping criteria is finally met.

$$\tau_{ijk} \leftarrow (1-\rho)\tau_{ijk} + \sum_{k=1}^{K+1} \Delta\tau_{ijk} \quad \forall (i,j,k) \tag{14}$$

$$\text{where} \quad \Delta\tau_{ijk} = \begin{cases} \delta/Cost & \text{if arc}(i,j,k)\text{is used by the best ant} \\ 0 & \text{Otherwise} \end{cases} \tag{15}$$

ACO solution quality is best known to be improved when a local search algorithm is embedded in the algorithm. We generate local neighboring solutions around the solutions found after each iteration and found that this step improved the solution significantly. The implemented local search algorithm was as straightforward as perturbing a small percentage we called it Local Search Ratio (LSR) of the values in B matrix.

Table 1
Summary of fit of the model.

R^2	0.985
R^2 Adj	0.89
Root mean square error	89.285
Mean of response	470.69
Observations (or sum wgts)	32

Table 2
Analysis of variance of the ACO results.

Source	DF	Sum of squares	Mean square	*F* ratio
Model	27	2064651.7	76468.6	9.5922
Error	4	31887.8	7972	**Prob > F**
C. total	31	2096539.5		0.02

4. Numerical experiments

Design of Experiments (DoE) was utilized to determine the appropriate values for the ACO parameters that will minimize the cost function. The factors considered along with their levels of low, medium, and high respectively are as follows: *No_of_Ants*: (5, 33, 60), ρ: (0.01, 0.155, 0.3), δ: (0.01, 0.155, 0.3), α: (1, 3, 5), β: (1, 3, 5), and τ_0: (0.5, 5.25, 10). The values of the parameter levels were selected based on many runs under different settings. To reduce the number of runs but reach sound conclusions, D-Optimal design was utilized, which has been shown to be an effective design (NIST/SEMATECH e-Handbook of Statistical Methods. Accessed February, 2010). JMP 6.0 from SAS was used to generate a D-Optimal design, with 32 experiments. The factors along with their interactions were analyzed using regression, ANOVA, and factors' effect tests. Three-factor interactions and higher were not considered as they typically have weak effect (Ross, 1996). Tables 1 and 2 show that the overall model fit and factor selection is meaningful with high R^2 value and low *p*-value for the overall model.

Following this, the significant factors were determined from the regression and based on a 95% confidence interval, a relatively large *t*-stat, and a small *p*-value (less than 0.05), a prediction expression was generated and solved for the minimum costs while varying the factors' values. As a result, the parameter values in Table 3 were determined to provide the best performance for the ACO. Note that the values of the parameters *LSR* and *Number of Iterations* were fixed for all experiments.

The problems we solve for are as follows:

Set 1 (5 5 3): $I = 5$ (origins), $J = 5$ (destinations), $K = 3$ (cross-docking facilities)
Set 2 (20 20 12): $I = 20$ (origins), $J = 20$ (destinations), $K = 12$ (cross-docking facilities)
Set 3 (50 50 30): $I = 50$ (origins), $J = 50$ (destinations), $K = 30$ (cross-docking facilities)
Set 4 (75 75 50): $I = 75$ (origins), $J = 75$ (destinations), $K = 50$ (cross-docking facilities)

Each problem category contains 10 problem instances.[1] The number of all possible enumerations in order to find the optimal solution for a problem instance can be expressed as follows: $(I \times J)^{K+1}$. This indicates that if we were to conduct exhaustive searches for the above problems (one problem instance each), then we would need the following number of iterations for the four sets (respectively): 3.91×10^5, 6.71×10^{33}, 2.17×10^{105}, and 1.80×10^{191}. Since major companies like 'Wal-Mart' assign the same sized truck on the most of its distribution routes, we assume that all the trucks are identical and have a same capacity of 24 items ($C = 24$). For the (5 5 3) problem set, optimal solutions were reached using B&B. Our algorithm was able to find the optimal solution except for one case as it is seen in Fig. 3. We solved the rest of the problems ((20 20 12), (50 50 30), (75 75 50)) by finding the lower bounds (LB), Branch-and-Bound (B&B) solutions for 30 min runs for each problem instance (using LINGO solver from Lindo Systems), and using our proposed algorithm. Note that as B&B was taking too much time, 30 min runs were used as limits.

ACO attained better solutions than B&B for large problems (sets 2, 3 and 4) in all instances. The averages of the found costs using ACO and B&B are summarized in Table 4. Figs. 4–6 reports the performance of ACO to solve problem sets 3, 4, and 5 in reference to the solution obtained from B&B. The *y*-axis in those figures is defined as the percentage of the difference between the obtained solution and the lower bound; i.e. $\frac{Cost(ACO)-LB}{LB} \times 100\%$ and $\frac{Cost(B\&B)-LB}{LB} \times 100\%$. You can see that ACO outperforms B&B for large problems and the difference between the two algorithms increases as the problem size increases (Table 4).

In order to ensure the superiority of the ACO solutions, we ran B&B for 1 h, 2 h, and 3 h, and compared the solutions. While B&B solutions did slightly improve, ACO still outperformed B&B in all problem instances. Fig. 7 depicts the percent deviation of B&B from ACO after running the previous for 30 min and 2 h. It is important to note here that the improvement of the B&B solution diminishes

Table 3
ACO parameters for all the problem instances.

Parameter	Value
Number of ants (M)	10
Evaporation rate (ρ)	0.1366
α	1
β	1.5
δ	0.3
Initial deposited pheromones (τ_0)	0.50
Local Search Ratio (*LSR*)	0.20
Number of iterations	5000

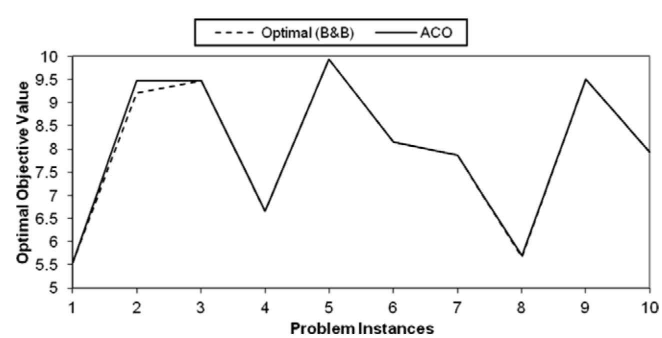

Fig. 3. Deviation between the optimal and ACO solutions for Set 1. ACO algorithm was able to find the optimal solutions for nine problem instances out of 10.

Table 4
Averages of the found costs using ACO and B&B for Sets 2, 3, and 4.

Problem	B&B	ACO	Improvement (%)
(20 20 12)	79.2814	76.7753	3.26
(50 50 30)	333.8496	315.5438	5.80
(75 75 50)	588.3218	541.5346	8.64

[1] The shipping quantities (S_{ij}) and cost parameters (c_{ij}, c_{ik}, c_{kj}) are summarized in the following site: http://filebox.vt.edu/users/rmusa/Cross_Dock_Problems/Problem_Instances.zip.

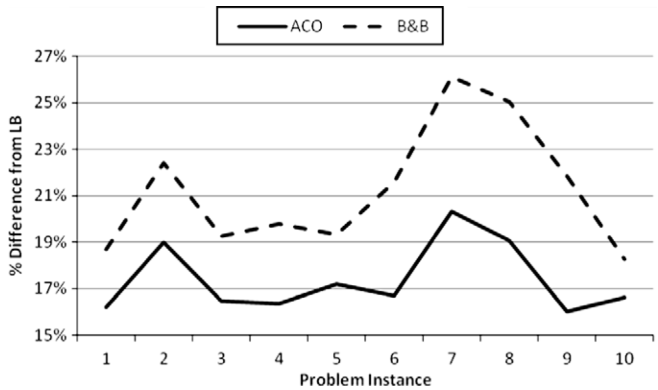

Fig. 4. Deviation between the optimal and ACO solutions from the lower bounds for Set 2. Notice that ACO outperforms B&B in all the instances.

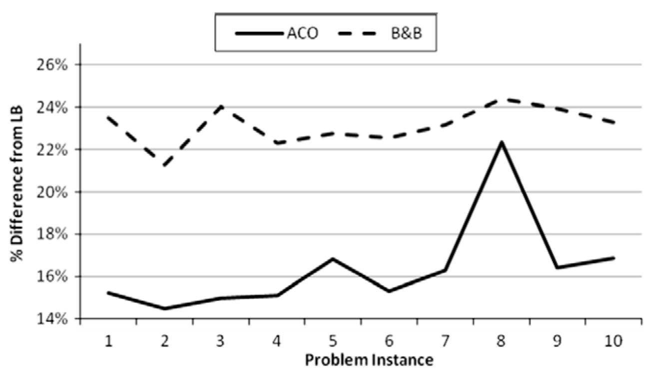

Fig. 5. Deviation between the optimal and ACO solutions from the lower bounds for Set 3. Notice that ACO outperforms B&B in all the instances.

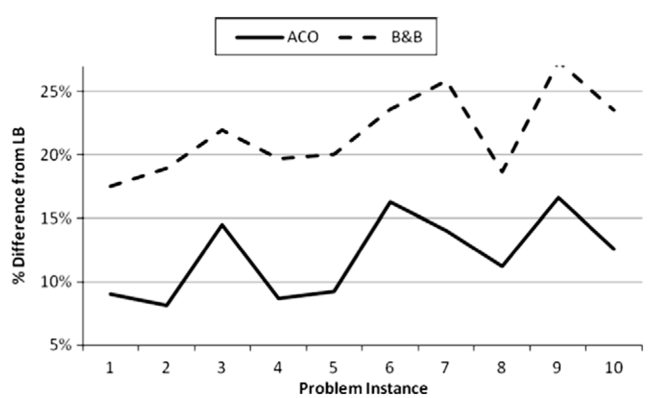

Fig. 6. Deviation between the optimal and ACO solutions from the lower bounds for Set 4. Notice that ACO outperforms B&B in all the instances.

with time. The averages of the costs for all problems using B&B running for 0.5, 1, 2, and 3 h are shown in Table 5. It can be seen that the B&B solution's improvement is very small when the latter is run for more than 1 h; in particular, the improvement from 2 to 3 h was less than 1%.

Figs. 8–11 depict sample results we obtained using our ACO algorithm for the third problem in Set 2. Fig. 8 shows the costs associated with the best and worst ants as the run progresses. The average cost was found to be non-increasing which indicates that the probabilistic search is directional rather than random. The maximum and minimum costs are found to converge with

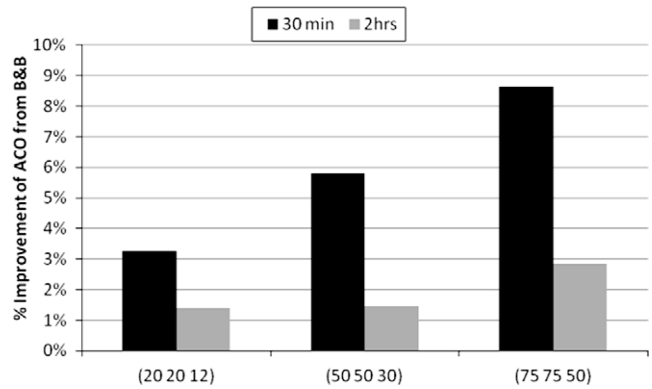

Fig. 7. Percent deviation between ACO and B&B solutions for 30 min and 2 h.

time and eventually meet at around iteration 3800. Furthermore, we can notice from Fig. 8 that the shipping costs were reduced from 200 units (arbitrary transportation plan) to 75 units (proposed algorithm); i.e. a 62.5% improvement. Since the objective of this IP was to collectively find the best route with the minimum number of trucks, we can see that the total number of trucks decrease with time (Fig. 9). In fact, the number of trucks was reduced from 560 to 430 (i.e. 30.2%). The number of trucks over time is not necessarily non-increasing since there are some instances where more trucks with good routes could result in a lower cost than fewer trucks with worse routes. Figs. 10 and 11 report the progress and contribution of each cost term in the objective function (1) and the number of trucks that are sent directly and indirectly in the network.

5. Concluding remarks and future research

We proposed a novel algorithm in this paper to solve the transportation problem of the cross-docking network. The objective is to minimize the total shipping cost of transporting pallets from a set of I suppliers to a set of J customers through K cross-docking distribution centers. This is to be performed with almost no storage in the distribution centers. Ant colony optimization (ACO) algorithm was used to solve the problem on hand. We solved a numerical example for verification and demonstrative purposes and found that our proposed solution finds solutions that significantly reduce the shipping cost in the network of cross-docks. The algorithm also outperformed Branch-and-Bound (B&B) solutions especially for large problem instances.

The authors are looking forward to extending their research work by relaxing some assumptions to make the problem and its solution more appropriate and applicable for several service sectors. Some of the ideas to be pursued are:

- Solve the transportation problem for the cross-docking network by taking the moving time into consideration. This will allow entering the service level as a constraining metric in the formulation.
- Solve a location-allocation problem for cross-docking facilities so the transportation cost will be minimized for third-party-logistics (3PL) companies.
- In our formulation, we assumed that the loads to be transported from origins to destinations are known. We could alternatively formulate the problem by determining the load amount to be transported from the origins to the destinations according to their demand. The load can be transported from different origins and can be also split in more than one truck at the origin.

Table 5
Averages of the found costs using B&B for Sets 2, 3, and 4.

Problem	B&B (30 min)	B&B (1 h)	B&B (2 h)	B&B (3 h)	Improvement % of B&B		
					½–1 h	1–2 h	2–3 h
(20 20 12)	79.2814	78.5307	77.8464	77.2891	0.956	0.879	0.721
(50 50 30)	333.8496	326.8754	320.1117	318.9159	2.134	2.113	0.375
(75 75 50)	588.3218	570.7967	557.0278	551.6581	3.070	2.472	0.973

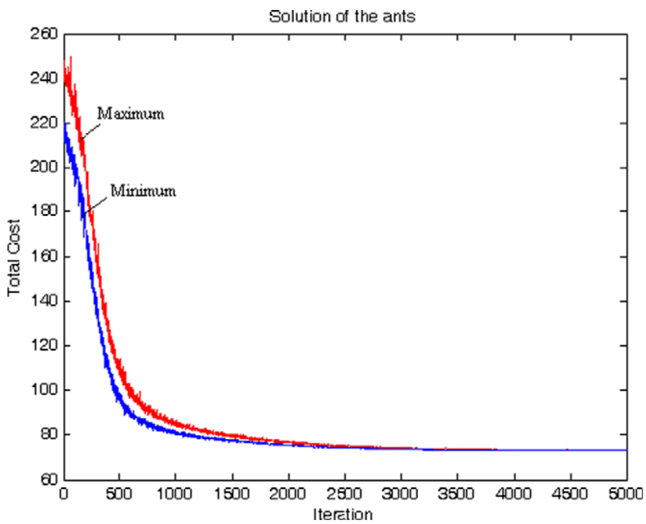

Fig. 8. Total costs for the best and worst ants over the iterations for the third problem instance in Set 2. The decreasing trend of the mean cost indicates that the search is directional. The best obtained solution at iteration 5000 for that problem was 73.238.

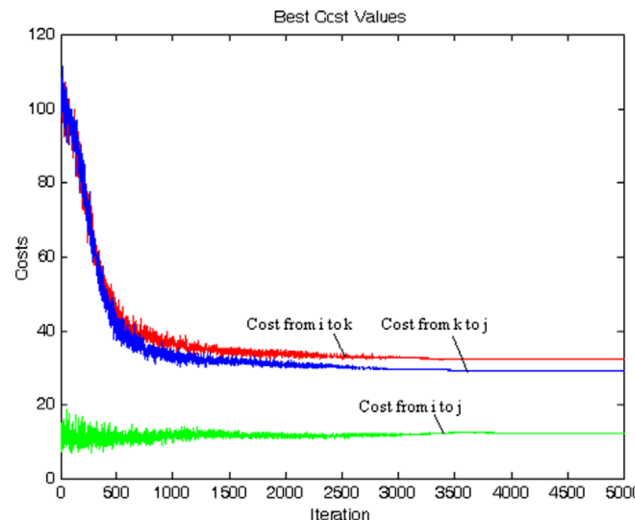

Fig. 10. Cost breakdown for the best ant over iterations for the third problem instance in Set 2.

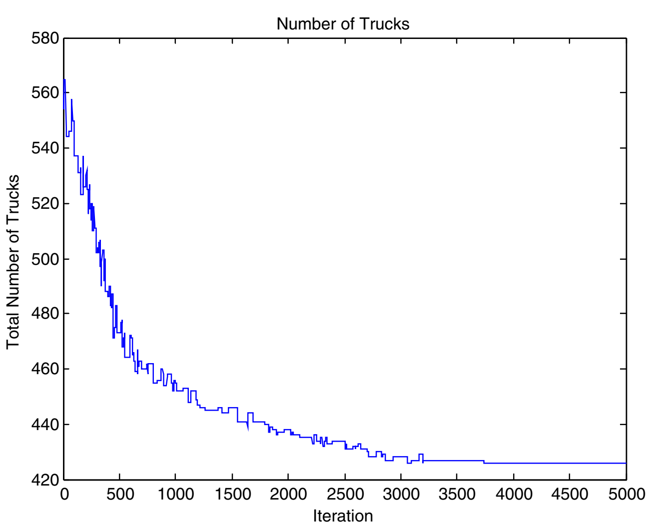

Fig. 9. Number of trucks found for the best ant over the iterations for the third problem instance in Set 2. Notice that the behavior here is not necessarily non-increasing because a lower cost with more trucks could be achieved for an optimized route.

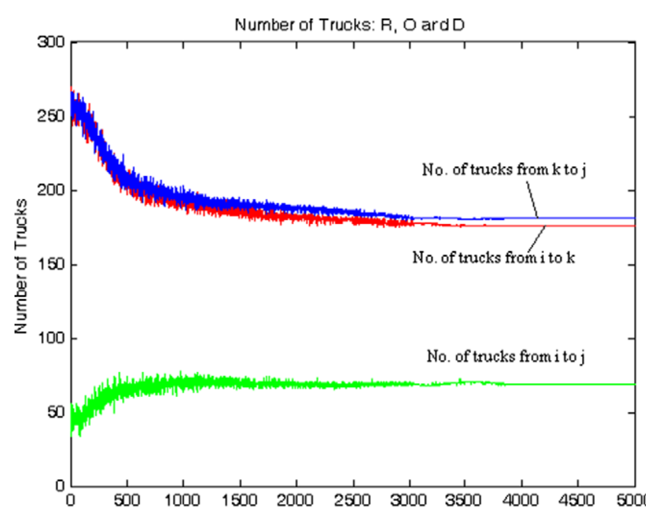

Fig. 11. Number of trucks breakdown for the best ant over iterations for the third problem instance in Set 2.

- Consider the availability of trucks in terms of their numbers and capacities. The authors will be working on relaxing this assumption through consideration of multiple period optimization and simulation.

Acknowledgement

The authors would like to sincerely thank Dr. Ahmed Ghoniem (assistant Professor in Isenberg School of Management – University of Massachusetts at Amherst) for his assistance and advice on the mathematical model.

References

Apte, U. M., & Viswanathan, S. (2000). Effective cross-docking for improving distribution efficiencies. *International Journal of Logistics, 3*, 291–302.

Bartholdi, J.J., Gue, K.R., & Kang, K. (2001). Staging freight in a crossdock. In *Proceedings of the international conference on industrial engineering and production management*.

Chen, P., Guo, Y., Lim, A., & Rodrigues, B. (2006). Multiple crossdocks with inventory and time windows. *Computers and Operations Research, 33*, 43–63.

Donaldson, H., Johnson, E.L., Ratliff, H.D., & Zhang, M. (1999). Schedule-driven cross-docking networks. Technical report, Georgia Institute of Technology. <http://www.isye.gatech.edu/apps/research-papers/papers/misc9904.pdf> Accessed 21.04.09.

Dorigo, M., Maniezzo, V., & Colorni, A. (2006). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, 26*, 29–41.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. The Massachusetts Institute of Technology Press.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP completeness*. W.H. Freeman.

Gue, K.R., Kang, K. (2001). Staging queues in material handling and transportation systems. In *Proceedings of the 2001 winter simulation conference*.

Li, Y., Lim, A., & Rodrigues, B. (2004). Crossdocking-JIT scheduling with time windows. *Journal of the Operational Research Society, 55*, 1342–1351.

NIST/SEMATECH e-Handbook of statistical methods. <http://www.itl.nist.gov/div898/handbook/> Accessed February, 2010

Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: Algorithms and complexity*. Dover Publications.

Pardalos, P. M., & Resende, M. G. C. (2002). *Handbook of applied optimization*. Oxford University Press.

Pardalos, P. M., & Romeijn, E. (2002). *Handbook of global optimization heuristics approaches* (Vol. 2). Kluwer Academic Publishers.

Ratliff, H.D., Vate, J.V., Zhang, M. (1999). Network design for load-driven cross-docking systems. Technical report, Georgia Institute of Technology. <http://www.isye.gatech.edu/apps/research-papers/papers/misc9914.pdf> Accessed 21.04.09.

Ross, P. (1996). *Taguchi techniques for quality engineering*. NewYork: McGraw Hill.

Tsui, L. Y., & Chang, C. H. (1992). An optimal solution to a dock door assignment problem. *Computers and Industrial Engineering, 23*, 283–286.