Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 51 (2006) 247–256
computers & industrial engineering
www.elsevier.com/locate/dsw

Vehicle routing scheduling for cross-docking in the supply chain

Young Hae Lee [a,*], Jung Woo Jung [a], Kyong Min Lee [b]

[a] *Department of Industrial Engineering, Hanyang University, 1271, Sa-1 Dong, Ansan, Gyeonggi Do 426-791, South Korea*
[b] *LG Phillips LCD, 642-3, Jinpyung Dong, Gumi, Gyeongsangbuk Do 730-726, South Korea*

Available online 25 September 2006

Abstract

One of the most important factors in implementing supply chain management is to efficiently control the physical flow of the supply chain. Due to its importance, many companies are trying to develop efficient methods to increase customer satisfaction and reduce costs. In various methods, cross-docking is considered a good method to reduce inventory and improve responsiveness to various customer demands. However, previous studies have dealt mostly with the conceptual advantages of cross-docking or actual issues from the strategic viewpoint. It is also necessary, however, to considering cross-docking from an operational viewpoint in order to find the optimal vehicle routing schedule. Thus, an integrated model considering both cross-docking and vehicle routing scheduling is treated in this study. Since this problem is known as NP-hard, a heuristic algorithm based on a tabu search algorithm is proposed. In the numerical example, our proposed algorithm found a good solution whose average percentage error was less than 5% within a reasonable amount of time.

Keywords: Supply chain management (SCM); Cross-docking; Vehicle routing scheduling; Tabu search algorithm

1. Introduction

Recently, many companies are being required to satisfy more complicated customer demands. Thus, many companies are trying to obtain a high level of agility, flexibility, and reliability for various demands. However, operations of a single company have a limit in improving customer satisfaction, because operations of a single company can have an adverse effect on those of the other companies in the supply chain. For that reason, supply chain management is attractive to many companies looking to efficiently improve customer satisfaction.

One of the most important things in implementing supply chain management is to efficiently control the physical flow of the supply chain. Apte and Viswanathan (2000) mentioned that 30% of price is incurred in the distribution process. Therefore, improvement of the material flow through efficient management of the distribution process is considered an essential activity to increase customer satisfaction. Thus, many companies are investigating and developing methods to efficiently control their material flow. In a number of these

* Corresponding author. Fax: +82 31 406 1089.
E-mail address: yhlee@hanyang.ac.kr (Y.H. Lee).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2006.02.006

methods, cross-docking is considered a good method to reduce inventory and improve customer satisfaction. This is why many companies choose cross-docking to manage the physical flow of their supply chains.

Cross-docking is defined as the continuous process to the final destination through the cross-dock, without storing products and materials in a distribution center (Apte & Viswanathan, 2000). When cross-docking is implemented in the supply chain, products in various locations are collected in the cross-dock prior to transportation to their destination. After classification according to product destination in the cross-dock, products are moved from the cross-dock to their respective destinations. This material flow in which cross-docking is implemented is illustrated in Fig. 1.

Two key points of cross-docking are simultaneous arrival and consolidation. In Fig. 1, all vehicles from suppliers arrive at the cross-dock simultaneously. If all vehicles do not arrive at the cross-dock simultaneously, some vehicles have to wait. Therefore, the timing for all vehicles in the pickup process has to be synchronized to reduce waiting time. According to the destination, all products are classified and loaded to each vehicle in the cross-dock. This is called consolidation. Then, all vehicles leave the cross-dock to distribute products to their destinations. If simultaneous arrival and consolidation can be easily accomplished in a supply chain's physical flow, all products can be moved from suppliers to customers without any interruptions. Therefore, we can expect to reduce inventory level and lead-time for delivery.

Generally, there is no inventory in the cross-dock. If stockout occurs frequently due to demand fluctuation and limited cross-dock capacity, a cross-docking system cannot be operated. In this case, some inventory must be stored in warehouses to respond to the uncertainty of future demands. However, we can more accurately forecast demands if the demands for the products in question are stable. For this reason, we maintain that the cross-docking system is best suited for products whose demand is stable and whose unit stockout cost is low. For example, groceries or agricultural products are typical products with stable demands and low unit stockout cost. It is necessary to quickly deliver these products to customers for freshness and because of the short period of circulation. Moreover, unit stockout cost for these products is relatively low. Thus, these products are appropriate for delivery through cross-docking.

To effectively apply cross-docking, both the pickup and delivery processes must be considered. The physical flow from the supplier to the cross-dock is called the pickup process. The core issue in the pickup process is simultaneous arrival at the cross-dock. Thus, we have to consider the vehicle routing and scheduling for simultaneous arrival. In the cross-dock, arrived products are classified into a certain group according to their destination. These products are then delivered to customers without delay or storage. Therefore, the number of arrived products at the cross-dock has to be equal to that of products delivered from the cross-dock. The process from the cross-dock to the customers is called the delivery process. As with the pickup process, the issue of vehicle routing scheduling must be considered. Thus, improvement of physical flow in the supply chain can be realized by the synthetic optimization of all processes, including pickup, cross-docking, and delivery. Moreover, routes and schedules of vehicles must be considered in order to synthetically optimize the physical flow. Therefore, we treat the vehicle routing/scheduling problem together with cross-docking to improve material flow in the supply chain.

Although the importance of cross-docking in a supply chain is widespread among companies, the many studies related to vehicle routing scheduling which consider the cross-docking system have not yet been treated. Most studies have focused on the concept of cross-docking and cases in point. Allen (2001) and Luton (2003) described the advantages of cross-docking. Apte and Viswanathan (2002) introduced cross-docking

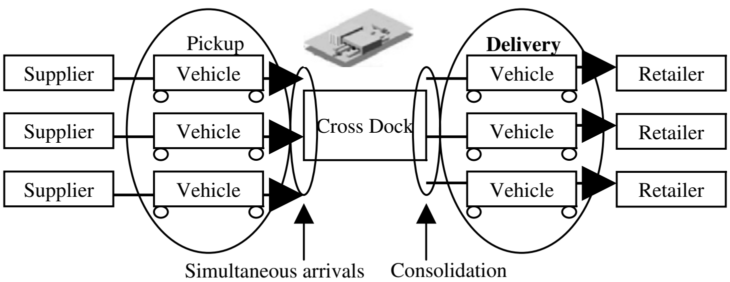

Fig. 1. The concept of cross-docking.

as one of the recent strategic and technological innovations in the management of a manufacturing supply chain. Recently, some studies treated actual issues of cross-docking from a strategic viewpoint. Jayaraman and Ross (2003) dealt with the PLOT (production, logistics, outbound, and transportation) design problem, which incorporates cross-docking into a supply chain environment. However, their approach was to assign a cross-dock to a customer zone with minimum cost at each strategic and operational level. Sung and Song (2003) considered an integrated service network design problem to find the location of cross-docks centers and allocation of vehicles. Then, a heuristic algorithm based on a tabu search algorithm was proposed. Gümüs and Bookbinder (2004) treated the problem to determine transportation policies within the network and whether to operate each cross-dock. Subsequently, commercial software including LINGO and CPLEX was used to find solutions. In summary, previous studies focused on the determination of optimal locations for cross-docks and the allocation of vehicles. Here, we treat the vehicle routing scheduling problem with a cross-docking system from an operational viewpoint. Therefore, we review previous studies on various vehicle routing problems in supply chain management.

There are huge studies which treat the vehicle routing problem in the supply chain. But, there is a close relationship between the pickup and delivery problem and the vehicle routing problem in cross-docking. The VRPTW (vehicle routing problem with time windows) can be especially helpful in treating the former problem, because one key point is simultaneous arrival at the cross-dock. Mosheiov (1998) handled the pickup and delivery problem which is a kind of vehicle routing problem. He developed the mathematical model to minimize transportation cost and maximize the efficiency of vehicles. Afterwords, two heuristic algorithms were proposed to find a good solution in a reasonable amount of time. Barbarosoglu and Ozgur (1999) reported that optimal transportation planning with multiple delivery centers in the supply chain can be replaced with multiple sub-optimizations, which means the optimization of transportation planning with one delivery center, because vehicles allocated to a certain distribution center take charge of an exclusive area. Moreover, they proposed a heuristic algorithm based on a neighborhood algorithm and a tabu search algorithm. Ceder, Golany, and Tal (2001) presented the scheduling problem for the simultaneous arrival of buses. If someone wants to change buses at a bus stop, a shorter waiting time is preferred. Therefore, this study tried to find the optimal schedule of buses to minimize waiting time at a bus stop. Hwang (2002) sought to determine the location of warehouses and the optimal schedule of vehicles by considering service level in the supply chain using a genetic algorithm. Lau, Sim, and Teo (2003) suggested a tabu search algorithm to minimize transportation costs for vehicle routing with specified time windows and a finite number of vehicles. In addition to the above studies, Rusdiansyah and Tsao (2005) presented the inventory routing problem encountered in vending machine supply chains under a vendor-managed inventory scheme. They referred to the existing periodic VRPTW to build their model.

In many studies, cross-docking was considered a good method to improve physical flow in the supply chain. However, most studies focused on the advantages of cross-docking. Recently, there have been studies on actual issues of cross-docking to improve material flow in the supply chain. However, these studies mostly treated the network design problem, i.e., the determination of appropriate locations of cross-docks. These studies are included in the category of strategic planning. Operational issues such as detailed vehicle routing scheduling with cross-docks were not dealt with. However, the integration of cross-docking with vehicle routing scheduling is important because this problem can be common in actual operations. This is why a model integrating cross-docking with vehicle routing scheduling is treated in this study. We suggest a mathematical model to determine the optimal vehicle routing schedule with a cross-dock to minimize the total cost. Because this problem is known as NP-hard, a new heuristic algorithm based on a tabu search algorithm is developed in this study.

This paper is organized as follows. In Section 2, the problem description and the mathematical model are presented. The proposed algorithm based on a tabu search algorithm is explained in Section 3. A numerical example is given in Section 4, and our conclusions are represented in Section 5.

2. Problem description

A cross-dock which takes charge of a specific area is assumed. Barbarosoglu and Ozgur (1999) stated that optimization of the entire distribution network can be achieved through optimization of each subset of the distribution network. Thus, a distribution network with a cross-dock is considered. Moreover, each vehicle is assumed to be allocated to this specific cross-dock, and split deliveries are not allowed. This is depicted in Fig. 2.

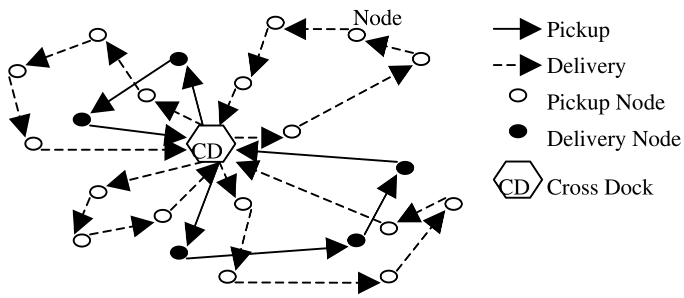

Fig. 2. A proposed network for a cross-dock.

P set of nodes in the pickup process
D set of nodes in the delivery process
0 cross-dock
n number of nodes (manufacturers or retailers)
m number of available vehicles
Q maximum capacity of the vehicle
p_i loading quantity in the pickup node i
d_i unloading quantity in the delivery node i
tc_{ij} transportation cost from node i to node j
c_k operational cost of the vehicle k
y_{ij} transported quantity of products from node i to node j in the pickup process
z_{ij} transported quantity of products from node i to node j in the delivery process
t_i length of a visit for the vehicle in node i
et_{ij} time for the vehicle to move from node i to node j
DT_i^k departure time of vehicle k from node i
AT^k arrival time of vehicle k at a cross-dock (ending time of the pickup process)

Decision Variable

$$x_{ij}^k : \begin{cases} 1, & \text{if the vehicle } k \text{ moves from the node } i \text{ to the node } j \\ 0, & \text{otherwise} \end{cases}$$

The objective of this problem is to determine the number of vehicles and the best route, schedule, and arrival time of each vehicle at a cross-dock to minimize the transportation cost, considering cross-docking in the planning horizon T. Moreover, we consider various constraints of vehicle departure and arrival in the distribution network, product quantity which can be loaded in a vehicle, and synchronized schedule of vehicles which arrive at the cross-dock. The mathematical model for our problem can be described as follows.

$$\text{Min} \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{m} tc_{ij} x_{ij}^k + \sum_{k=1}^{m} \sum_{j=1}^{n} c_k x_{0j}^k \tag{1}$$

s.t

$$\sum_{i=0}^{n} \sum_{k=1}^{m} x_{ij}^k = 1, \quad \forall j \tag{2}$$

$$\sum_{j=0}^{n} \sum_{k=1}^{m} x_{ij}^k = 1, \quad \forall i \tag{3}$$

$$\sum_{i=0}^{n} x_{ip}^k - \sum_{j=0}^{n} x_{pj}^k = 0, \quad \forall p, k \tag{4}$$

$$\sum_{j=1}^{n} x_{0j}^{k} \leqslant 1, \quad \forall k \tag{5}$$

$$\sum_{i=1}^{n} x_{i0}^{k} \leqslant 1, \quad \forall k \tag{6}$$

$$\sum_{k=1}^{m} \sum_{j=1}^{n} x_{0j}^{k} \leqslant m, \quad \forall k \tag{7}$$

$$y_{ij} + z_{ij} \leqslant Q \cdot x_{ij}^{k}, \quad \forall k, \forall (i,j) \tag{8}$$

$$\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} d_i \tag{9}$$

$$y_{jl} - y_{ij} = \begin{cases} p_j, & \text{if } j \in P, \forall i, l \\ 0, & \text{if } j \in D, \forall i, l \\ -\sum_{i=1}^{n} p_i, & \text{if } j \in 0, \forall i, l \end{cases} \tag{10}$$

$$z_{ij} - z_{jl} = \begin{cases} 0, & \text{if } j \in P, \forall i, l \\ d_j, & \text{if } j \in D, \forall i, l \\ \sum_{i=1}^{n} d_i, & \text{if } j \in 0, \forall i, l \end{cases} \tag{11}$$

$$\sum_{i=0}^{n} \sum_{j=0}^{n} t_i^k x_{ij}^k + \sum_{i=0}^{n} \sum_{j=0}^{n} et_{ij}^k x_{ij}^k \leqslant T, \quad \forall k \tag{12}$$

$$DT_j^k = (et_{ij} + DT_i^k + t_j) x_{ij}^k, \quad \forall k \tag{13}$$

$$AT^k = (DT_i^k + et_{i0}) x_{i0}^k, \quad \forall k \tag{14}$$

$$AT^k = AT^{k'}, \quad \forall k \neq k' \tag{15}$$

The objective equation of this problem is shown as Eq. (1). This is the sum of transportation cost and fixed cost of vehicles, and we want to minimize this cost. Eqs. (2) and (3) show that one vehicle has to arrive at and leave one node. Eq. (4) expresses the consecutive movement of vehicles. Whether or not a vehicle arrives at and leaves a cross-dock is shown in Eqs. (5) and (6). Eq. (7) shows that the number of vehicles that leave a cross-dock must be less than the number of available vehicles m. Eq. (8) expresses that the quantity of loaded products in a certain vehicle can not exceed the maximum capacity of the vehicle. The flow conservation for products is manifested in Eq. (9). The quantity of products en route (between nodes) in the pickup and delivery process is shown in Eqs. (10) and (11). In Eq. (12), the sum of the total length of the visit to each node and total transportation time must be less than the planning horizon T. Eq. (13) expresses that the departure time of a vehicle from a node is determined by the sum of the arrival time at a node, the length of a visit, and time to move. The arrival time at a cross-dock is represented in Eq. (14), and the constraint for simultaneous arrival to a cross-dock is given in Eq. (15).

3. Application of a tabu search algorithm

The pickup and delivery problem, can be considered as a kind of the vehicle routing problem (Mosheiov, 1998). This problem can then be classified as an NP-hard problem. Therefore, an efficient heuristic algorithm is needed to obtain a good solution in a reasonable amount of time. A tabu search algorithm is one of various meta-heuristic algorithms. According to Barbarosoglu and Ozgur (1999), a tabu search algorithm is the most suitable meta-heuristic algorithm, because it is possible to escape the local solution by the creation of various neighborhoods and to prevent return to the local solution, and has a memory of the search processes used.

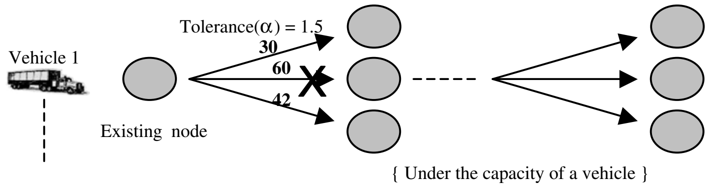

Fig. 3. An example of generating an initial solution.

Baker and Sheasby (1999) stated that a tabu search algorithm can be the most effective in optimizing the vehicle routing problem. Thus, a tabu search algorithm was used in this study to find an efficient operational strategy of vehicles.

The initial solution of our proposed algorithm was generated by the procedure described in Fig. 3. For the first vehicle, all available routes by which this vehicle can leave the existing node are searched. Then, a candidate list is made up of the routes for which the ratio of transportation cost to minimum transportation cost is less than α. Then, a randomly selected route and its related node in the candidate list are allocated to the vehicle. This is continued until the capacity of a vehicle is reached, and the same procedure is executed for all remaining vehicles. Fig. 3 describes the procedure for allocating the routes for vehicle 1 where the tolerance α is equal to 1.5. From the existing node, three available nodes are found, and two nodes are selected because their ratio of transportation cost to minimum transportation cost (30) is less than the tolerance of 1.5. Then, a randomly selected node in the two routes is allocated for vehicle 1.

Fig. 4 shows an example of the generation of an adjacent solution. For vehicle 1, two routes whose cost of transportation between two nodes is more than the other routes are found. Then, the two selected routes are exchanged with corresponding routes for another randomly selected vehicle 2. If there are not any corresponding routes or additional vehicle capacity is needed, the sequence of the two routes is changed. This is continued for all vehicles.

Then, our proposed algorithm can be summarized as the following:

Step 1: Initialization
  Initialize α to generate solutions.
Step 2: Generation of an initial solution
  2.1 Pickup process
    2.1.1 Select a vehicle to route.
    2.1.2 Search all available routes from the existing node. Generate the candidate list made up of routes whose ratio of transportation cost from the existing location has a minimum transportation cost of less than α.
    2.1.3 Select a route and its related node randomly from the candidate list.
    2.1.4 Allocate a product in the selected node to a selected vehicle, and replicate 2.1.2 and 2.1.3 under the capacity of a vehicle. If the capacity of a vehicle is exceed, execute 2.1.5.
    2.1.5 Select another vehicle to route. Replicate 2.1.2–2.1.4 for all remaining vehicles.
  2.2 Delivery process
    Replicate the same steps in the pickup process.

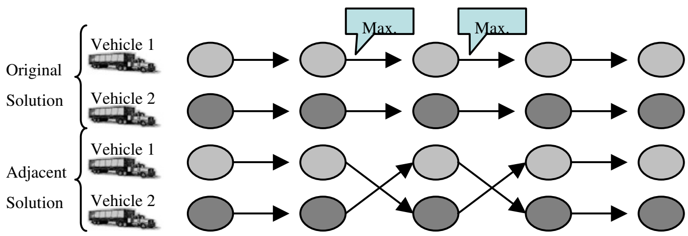

Fig. 4. An example of generating an adjacent solution if there are corresponding routes.

Step 3: Tabu search algorithm
3.1 Generation of an adjacent solution
3.1.1 Pickup process
3.1.1.1 Find two routes whose transportation cost between two nodes is more than the other nodes.
3.1.1.2 Exchange the two routes with the routes in corresponding routes for another randomly selected vehicle. If there are not any corresponding routes or additional vehicle capacity is needed, the sequence of the two routes is changed.
3.1.1.3 If the total cost is improved, keep the present solution as the best solution. If not, execute 3.1.1.1.
3.1.1.4 Find the best solution by repeating 3.1.1.1 $\sim$ 3.1.1.3.
3.1.2 Delivery process
Replicate the same steps as in the pickup process.
3.2 Generation of the tabu list
Found pairs (route, vehicle), which are the result in 3.1.1.4, are added to the tabu list.
3.3 Terminating condition 1
If the limited number of tabu searches is not exceeded, replicate Step 3.
Step 4: Terminating condition 2
If the limited number of generated initial solutions was not exceeded, replicate Steps 2 and 3.

4. Numerical example

In this section, we treat some numerical examples and report the performance of our proposed algorithm. Prior to describing the numerical example of our heuristic algorithm, we need to find an optimal solution. In this study, we obtain an optimal solution by enumeration, which finds the best of all possible solutions. This method does take more time than heuristic algorithms. For convenience, the length of a visit for a vehicle to node i, t_i, is assumed to be 0. It is also assumed that the planning horizon, T, is 16 h. In Table 1, the value of the parameters is illustrated. For randomly generated problems with 10, 30, and 50 vehicles, the resulting schedule for each generated problem by enumeration is shown in Tables 2–4, respectively. Each value of transportation costs between two nodes is generated asymmetrically. Table 2 shows the resulting schedule for 6 vehicles to visit 10 nodes. Five vehicles in the pickup process arrived at the cross-dock in 284 min simultaneously. It took 645 min to complete all processes. In Table 3, 17 vehicles were needed to route 50 nodes. Nine vehicles operated in the pickup process arrived at the cross-dock in 338 min simultaneously, and it took 731 min to conclude all processes. Table 4 shows the result of the problem with 50 nodes. Nine and 8 vehicles were input into the pickup process and the delivery process, respectively. Vehicles in the pickup process arrived at the cross-dock in 338 min, and it took 729 min to complete all processes.

To find a good solution to this problem in a reasonable amount of time, our developed heuristic algorithm was applied. The length of tabu tenure was assumed to be 3, and 1000 iterations were executed for a tabu search process. However, the tolerance value was a critical factor in the performance of our heuristic

Table 1
Parameter values

	Problem 1	Problem 2	Problem 3
n	10	30	50
m	10	20	30
T	960	960	960
Q	70	150	150
c_k	1000	1000	1000
et_{ij}	Uniform (20,200)	Uniform (20,100)	Uniform (20,200)
tc_{ij}	Uniform (48,560)	Uniform (48,480)	Uniform (48,560)
p_i, d_i	Uniform (5,50)	Uniform (5,20)	Uniform (5,30)
Number of pickup nodes	4	7	12
Number of delivery nodes	6	23	38

Table 2
The resulting schedule for each vehicle to visit 10 nodes

Vehicle	Visited node (starting time)
1	0 (68) → 2 (190) → 1 (234) → 0 (284)
2	0 (0) → 4 (156) → 0 (284)
3	0 (96) → 3 (184) → 0 (284)
4	0 (284) → 10 (377) → 6 (400) → 8 (494) → 0 (645)
5	0 (284) → 9 (432) → 7 (527) → 0 (591)
6	0 (284) → 5 (343) → 0 (393)

*Pickup nodes, 1–4; delivery nodes, 5–10; total cost, 10102; cross-docking time, 284 min.

Table 3
The resulting schedule for each vehicle to visit 30 nodes

Vehicle	Visited node (starting time)
1	0 (41) → 4 (62) → 2 (92) → 0 (181)
2	0 (0) → 7 (59) → 6 (99) → 0 (181)
3	0 (69) → 1 (116) → 0 (181)
4	0 (35) → 5 (114) → 0 (181)
5	0 (64) → 3 (128) → 0 (181)
6	0 (181) → 22 (202) → 16 (234) → 24 (256) → 26 (281) → 13 (321) → 18 (358) → 0 (381)
7	0 (181) → 29 (202) → 9 (224) → 14 (259) → 23 (300) → 10 (322) → 0 (356)
8	0 (181) → 30 (215) → 20 (238) → 12 (269) → 11 (304) → 8 (237) → 0 (356)
9	0 (181) → 19 (215) → 21 (262) → 17 (300) → 25 (324) → 0 (363)
10	0 (181) → 27 (233) → 28 (259) → 15 (281) → 0 (323)

*Pickup nodes, 1–7; delivery nodes, 8–30; total cost, 14373.6; cross-docking time, 181 min.

Table 4
The resulting schedule for each vehicle to visit 50 nodes

Vehicle	Visited node (starting time)
1	0 (116) → 7 (143) → 11 (189) → 0 (338)
2	0 (125) → 6 (245) → 0 (338)
3	0 (0) → 3 (195) → 5 (278) → 0 (338)
4	0 (171) → 2 (262) → 0 (338)
5	0 (201) → 9 (284) → 0 (338)
6	0 (127) → 1 (224) → 0 (338)
7	0 (73) → 8 (233) → 0 (338)
8	0 (181) → 12 (217) → 10 (247) → 0 (338)
9	0 (83) → 4 (245) → 0 (338)
10	0 (338) → 44 (364) → 45 (385) → 30 (410) → 31 (432) → 20 (457) → 15 (479) → 27 (518) → 22 (567) → 0 (632)
11	0 (338) → 16 (365) → 33 (388) → 17 (415) → 25 (447) → 0 (555)
12	0 (338) → 27 (359) → 35 (388) → 18 (409) → 41 (431) → 0 (570)
13	0 (338) → 39 (399) → 46 (428) → 32 (450) → 21 (472) → 14 (494) → 38 (631) → 0 (731)
14	0 (338) → 42 (504) → 26 (568) → 40 (589) → 29 (632) → 37 (674) → 0 (727)
15	0 (338) → 23 (441) → 34 (510) → 48 (536) → 50 (560) → 0 (582)
16	0 (338) → 36 (412) → 43 (451) → 28 (488) → 13 (532) → 0 (633)
17	0 (338) → 19 (474) → 49 (504) → 24 (533) → 0 (677)

*Pickup nodes, 1–12; delivery nodes, 13–50; total cost, 28253.2; cross-docking time, 338 min.

algorithm. In order to investigate the effect of the tolerance value, we experimented with three randomly generated problems equal to the above problems used to obtain an optimal solution with 10, 30, and 50 nodes. Then, the value of the average percentage error was reported from 30 repetitions. Percentage error was computed according to Eq. (16).

$$100 \times \frac{|\text{total cost of a heuristic solution} - \text{total cost of the optimal solution}|}{\text{total cost of the optimal solution}} \tag{16}$$

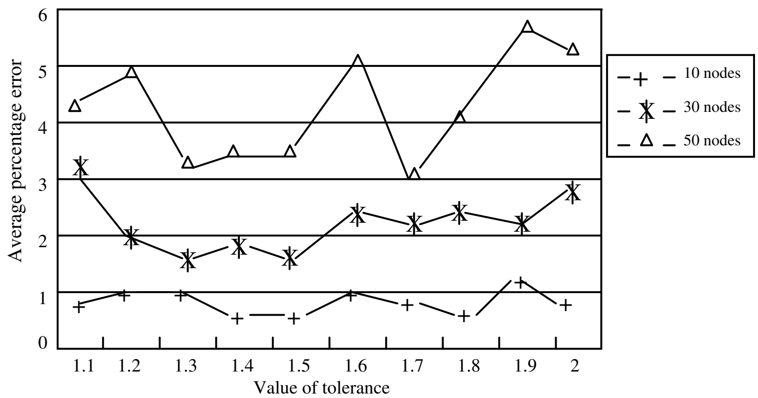

Fig. 5. Average percentage error vs. value of tolerance.

Fig. 5 shows the average percentage error with changing tolerance value. However, it is difficult to say that there is a certain relationship between average percentage error and tolerance value. That is, an increasing or decreasing value of tolerance can not guarantee an increasing or decreasing average percentage error.

Table 5
Comparison of the proposed algorithm and enumeration in 30 replications

Total cost	Proposed algorithm			Enumeration			Error/enumeration (%)		
Nodes: Times	10	30	50	10	30	50	10	30	50
1	8948.4	14972.8	28368	8948.4	14972.8	26883.6	0	0	5.52
2	9093.6	14625.6	30193.6	9093.6	14587.2	28250	0	0.26	6.88
3	9060	14846.8	29376	9060	14846.8	29022.8	0	0	1.22
4	7211.6	14768.4	29303	7211.6	14768.4	28303.6	0	0	3.53
5	7581.2	14575.2	29714.8	7581.2	14316.4	28897.2	0	1.81	2.83
6	9356.8	14603.2	30053.6	9356.8	14603.2	28560.8	0	0	5.23
7	7200.4	14720.8	28788	7200.4	14720.8	28454.4	0	0	1.17
8	7396.4	15126.8	30070.4	7396.4	14912.6	28076.4	0	1.44	7.1
9	7385.2	14734.8	29213.6	7385.2	14734.8	28563.6	0	0	2.28
10	9206	14636.8	28292.4	9206	14636.8	27160.8	0	0	4.17
11	10555.6	14942	30249.6	10555.6	14829.7	29428.8	0	0.76	2.79
12	7312.4	14606	30969.2	7312.4	14606	30744.8	0	0	0.73
13	6970.8	14967.7	29350.8	6970.8	14967.7	28328	0	0	3.61
14	9614.8	15070.8	29252.8	9614.8	14932.1	27883.6	0	0.93	4.91
15	8956.4	14858	27998.4	8956.4	14669.4	27695.2	0	1.29	1.09
16	9738	14762.8	29504.8	9738	14472.9	27110.4	0	2	8.83
17	8950.8	14659.2	28519.2	8950.8	14659.2	28214.4	0	0	1.08
18	6632	14681.6	28362.4	6632	14681.6	27628.4	0	0	2.66
19	7304	14768.4	28138.4	7304	14682.4	25111.2	0	0.59	12.05
20	7763.2	14712.4	27852.8	7763.2	14582.4	27419.6	0	0.89	1.58
21	6671.2	14180.4	28634	6671.2	14180.4	27320.8	0	0	4.8
22	7194.8	14586.4	29432	7194.8	14586.4	28905.6	0	0	1.82
23	6682.4	15110	30288.8	6682.4	14987.2	28846.8	0	0.82	5
24	7690.4	14499.6	28012.4	7690.4	14499.6	27886.4	0	0	0.45
25	8802.4	14659.2	28242	8802.4	14659.2	27960.4	0	0	1.01
26	10642.4	15191.2	28485.6	10642.4	14998.7	27910.8	0	1.28	2.06
27	8878	14340	30501.6	8878	14340	29731.6	0	0	2.59
28	6982	14544.4	29238.8	6982	14186.4	28602.4	0	2.52	2.22
29	10933.6	14412.8	29523.8	10933.6	14412.8	26802.4	0	0	10.15
30	7452.4	14244.8	28681.6	7452.4	14244.8	27681.6	0	0	3.61
Average error							0	0.49	3.77
Standard deviation of error							0	0.72	2.88
Maximum error							0	2.52	12.05

However, although we can obtain smaller average percentage error with tolerance values of 1.3, 1.4, and 1.5, it can not be said that these tolerance values are better than all other values. Moreover, we can obtain a solution which has an average percentage error less than 6% for the worst case in three problems. Therefore, the value of tolerance is set from 1.0 to 2.0, increasing by intervals of 0.1.

To find the effect of our proposed algorithm, we compared the result of our algorithm with that of enumeration through 30 replications for each problem. Table 5 shows that there was at most a 4% average error of total cost between our proposed algorithm and enumeration. Therefore, we can conclude that our proposed algorithm was able to efficiently find a good solution.

5. Conclusions

Cross-docking is considered an efficient method to control physical flow in the supply chain. This study treated a model integrating cross-docking with the pickup and delivery process in the supply chain. Then, a mathematical model was developed to determine an optimal vehicle routing schedule which considered cross-docking. Since this problem is known as NP-hard, an algorithm based on a tabu search algorithm was also developed. In 30 randomly generated problems, we found near optimal solutions whose average percentage error was less than 4% in a reasonable amount of time. Recently, the remanufacturing and recycling processes have also been considered important. Our problem in this study can easily be extended, and should be extended to the remanufacturing and recycling processes.

References

Allen, J. (2001). Cross docking: Is it right for you? *Canadian Transportation Logistics, 104*, 22–25.

Apte, U. M., & Viswanathan, S. (2000). Effective cross docking for improving distribution efficiencies. *International Journal of Logistics, 3*, 91–302.

Apte, U. M., & Viswanathan, S. (2002). Strategic and technological innovations in supply chain management. *International Journal of Manufacturing Technology and Management, 4*, 264–282.

Baker, B. M., & Sheasby, J. (1999). Extensions to the generalized assignment heuristic for vehicle routing. *European Journal of Operational Research, 119*, 147–157.

Barbarosoglu, G., & Ozgur, D. (1999). A Tabu search algorithm for the vehicle routing problem. *Computer & Operational Research, 26*, 255–270.

Ceder, A., Golany, B., & Tal, O. (2001). Creating bus timetables with maximal synchronization. *Transportation Research Part A, 35*, 913–928.

Gümüs, M., & Bookbinder, J. H. (2004). Cross-docking and its implications in location-distribution systems. *Journal of Business Logistics, 25*, 199–229.

Hwang, H. S. (2002). Design of supply chain logistics system considering service level. *Computers & Industrial Engineering, 43*, 283–297.

Jayaraman, V., & Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research, 144*, 629–645.

Lau, H. C., Sim, M., & Teo, K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research, 148*, 559–569.

Luton, D. (2003). Keep it moving: a cross-docking primer. *Materials Management and Distribution, 48*, 29–30.

Mosheiov, G. (1998). Vehicle routing with pick-up and delivery: tour – partitioning heuristics. *Computers & Industrial Engineering, 34*, 669–684.

Rusdiansyah, A., & Tsao, D. (2005). An integrated model of the periodic delivery problems for vending-machine supply chains. *Journal of Food Engineering, 70*, 421–434.

Sung, C. S., & Song, S. H. (2003). Integrated service network design for a cross-docking supply chain network. *Journal of the Operational Research Society, 54*, 1283–1295.