Expert Systems with Applications 37 (2010) 6868–6873

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Vehicle routing with cross-docking in the supply chain

Ching-Jong Liao [a,*], Yaoming Lin [a], Stephen C. Shih [b]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan
[b] School of Information Systems and Applied Technologies, Southern Illinois University, Carbondale, IL 62901, USA

ARTICLE INFO

Keywords:
Cross-docking
Supply chain management
Vehicle routing problem
Scheduling
Tabu search

ABSTRACT

Cross-docking is considered as an efficient method to control the inventory flow, which is essential in supply chain management. In this paper, we consider a model that integrates cross-docking into the vehicle routing problem. In the model, a set of identical vehicles are used to transport goods from supplies to retailers through a cross-dock and the whole process must be completed in the planning horizon. Each supplier and retailer can be visited only once and the total quantity of goods in a vehicle must be less than its capacity. The objective of the problem is to determine the number of vehicles and a set of vehicle schedules with a minimum sum of operational cost and transportation cost. A new tabu search (TS) algorithm is proposed to obtain a good feasible solution for the problem. Through extensive computational experiments, it is shown that the proposed TS algorithm can achieve better performance than an existing TS algorithm while using much less computation time. The average improvements are as high as 10–36% for different size of problems.

1. Introduction

In general, the production procedure consists of purchasing raw materials from suppliers, manufacturing, storing, and delivering end products to customers. The system involving moving products or services from suppliers to customers is referred to as a supply chain. Traditionally, members in a supply chain such as suppliers, manufacturers, and customers are looking for the highest efficiency simply for themselves, and typically do not consider global optimization or total efficiency by factoring in other members in the supply chain. Consequently, once the upstream and downstream sides cannot coordinate with each other in a supply chain, it will incur a higher system cost. In order to reduce the total cost in a supply chain, it is indispensable to consider all supply chain members at the same time and use more effective methods to achieve a lower system cost. This endeavor is considered an essential task of supply chain management and can be formally defined as a set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses, and stores. As a result, products can be produced and distributed at the right quantities, to the right location, and at the right time, which results in minimal system-wide costs while fulfilling customer demands (Simchi-Levi, Kaminsky, & Simchi-Levi, 2003).

According to the principles of accounting, inventory can have different names depending on where it resides at the various stages of the production life cycle. Inventory can be seen as raw materials when it is stored in the supplier's warehouse. It can be work-in-process (WIP) (or called in-process inventory) when the material is in the procedure of production. Furthermore, inventory can be finished goods when WIP is completed and dispatched to the customer. Raw materials, work-in-process, and finish goods are all inventory. One of the major concepts of supply chain management is to control the flow of inventory.

A warehousing strategy, called cross-docking, is considered a viable method to reduce inventory while satisfying customers' needs. Through streamlining the flow between the suppliers and manufacturers, this strategy can help diminish inventory storage. Cross-docking deals with movement of goods directly from the receiving dock to the shipping dock, where the goods are stored in a cross-dock for a short time, usually less than 12 h, or just directly dispatched to the customers (Apte & Viswanathan, 2000; Kreng & Chen, 2008). The cross-docking strategy essentially eliminates the inventory holding function of a traditional warehouse while still allowing products to be classified and loaded to the delivery vehicles through a consolidation process (Wen, Larsen, Clausen, Cordeau, & Laporte, 2008). The concept of cross-docking is depicted in Fig. 1 in which the two key points are simultaneous arrival and consolidation. If vehicles of the pickup fleet could not arrive at the cross-dock simultaneously, then the consolidation process would be delayed until all goods are collected, and thereby increasing the waiting time and the inventory level at the cross-docking.

* Corresponding author. Address: Department of Industrial Management, National Taiwan University of Science and Technology, 43 Keelung Road, Section 4, Taipei 106, Taiwan. Fax: +886 2 27376344.
E-mail address: cjliao@mail.ntust.edu.tw (C.-J. Liao).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.03.035

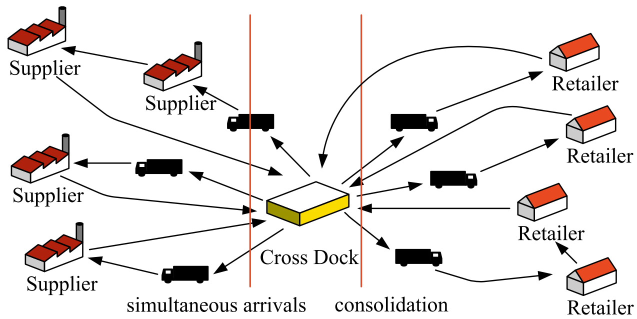

Fig. 1. The concept of cross-docking.

Most studies on cross-docking discuss the concept of cross-docking, its physical design, or location determination. Apte and Viswanathan (2000) proposed a framework for designing a cross-docking system and indicated that cross-docking can effectively bring substantial reduction in the transportation cost without increasing inventory. Sung and Song (2003) proposed a tabu search algorithm for an integrated service network to find the locations of both cross-docks and vehicles. They pointed out that the cross-docking strategy has been acknowledged as having great potential to reduce transportation cost and delivery time without increasing inventory. Gumus and Bookbinder (2004) used commercial software including LINDO and CPLEX to determine transportation policies in a logistic network and optimal locations of cross-docks. Recently, Kreng and Chen (2008) developed two models, a cross-docking model and a traditional warehousing model, to coordinate both production and distribution in order to reduce relevant costs in a supply chain.

On the other hand, the classic vehicle routing problem (VRP) involves the service of a set of customers with known demands by a fleet of vehicles from a single distribution center. The objective of the VRP is to minimize the total distance and the number of vehicles which start and end their tours at the central depot. Mosheiov (1998) stated that many applications of VRP involving pickup and delivery services are referred to the pickup and delivery problem (PDP). In a PDP, it is necessary to meet the needs of two special kinds of customers: demand customers and supply customers. For the demand customers, they need a shipment from a depot or other supply customers. For the supply customers, they need a shipment to take the goods away. The objective of the problem is to find a minimum length tour for a capacitated vehicle where the length tour starts at a depot loaded with enough goods to satisfy the customers, travels in the network to deliver the goods to the demand customers, and collects the goods from the supply customers (Tzoreff, Granot, Granot, & Sosic, 2002).

As discussed above, although there exist many studies on cross-docking and VRP, there are few papers that consider both cross-docking and VRP simultaneously. Dealing with a VRP task with cross-docking is important because the problem is quite common in practice. The work of Lee, Jung, and Lee (2006) is probably the first that takes both VRP and cross-docking into consideration. They proposed a tabu search (TS) to determine the number of vehicles and the optimal vehicle routing schedule at a cross-dock to minimize the sum of transportation cost and fixed cost of vehicles. In this paper, a new tabu search (TS) algorithm is developed and a comparison of its performance with Lee et al.'s TS is presented.

The remaining of this paper is organized as follows. The problem description is presented in Section 2. The proposed TS algorithm is given in Section 3. Section 4 contains all computational experiments, followed by the conclusions in Section 5.

2. Problem description

According to Barbarosoglu and Ozgur (1999), optimal transportation planning can be replaced by multiple sub-optimizations in supply chain management because vehicles allocated to a certain distribution center take charge of an exclusive area. Thus, a distribution network with only a cross-dock is considered in this paper. The considered problem is described by Fig. 2, where it is assumed that all the vehicles are located in the cross-dock and split deliveries are not allowed. In the figure, a triangle represents a supplier and a cycle represents a retailer. The pickup vehicles start from the cross-dock and arrive at the cross-dock simultaneously. Then, the delivery vehicles move to the retailers and return to the cross-dock after completing their tours. The objective of the problem is to determine the number of vehicles and the best route as well as the arrival time of each vehicle so as to minimize the sum of the operational cost of vehicles and the transportation cost.

The following notation is used throughout the paper:

n	number of nodes (suppliers or retailers) in the logistic network
m	number of available vehicles, which are all identical
Q	capacity of the vehicle, which is common to all vehicles
p_i	loading quantity in the pickup node i
d_i	unloading quantity in the delivery node i
c_{ij}	transportation cost from node i to node j
t_{ij}	travel time between node i and node j
o_k	operational cost of vehicle k
T	planning horizon

The limitations of the problem are described as follows:

1. The transportation time for the pickup and delivery process must be less than T minutes.
2. Each supplier or retailer can only be picked up or delivered once.
3. The pickup vehicles should arrive at the cross-dock simultaneously.
4. The number of vehicles utilized should be less than or equal to m.
5. The total quantity of pickup should equal the quantity to be delivered.
6. For each vehicle, the load on the pickup route and on the delivery route cannot exceed the capacity of the vehicle.

3. The tabu search approach

Since the considered problem is NP-hard (Lee et al., 2006), metaheuristics are used to search for a more feasible solution. Using the same approach as Lee et al. (2006), tabu search (TS) is employed as the solution method because it has proven to be one of the best available metaheuristics for solving VRPs (Cordeau, Gendreau, Laporte, Potvin, & Semet, 2002; Wen et al., 2008). In general, TS contains at least five elements: an initial solution, a neighborhood structure, stopping criteria, a tabu list, and aspiration criteria. Starting from an initial solution, TS moves from a solution x to another solution x' in the neighborhood of x until a certain stopping criterion is satisfied. The tabu list contains the solutions that have been visited in the recent past. In the TS move, it excludes solutions of recent visits in order to escape from a local optimum and cycling search. Furthermore, aspiration criteria are used to revoke a tabu list when there is an attractive tabu move.

In what follows, we propose a scheme to generate an initial solution to the considered problem. In the initial solution scheme, only a full truckload is considered in order to fully utilize the

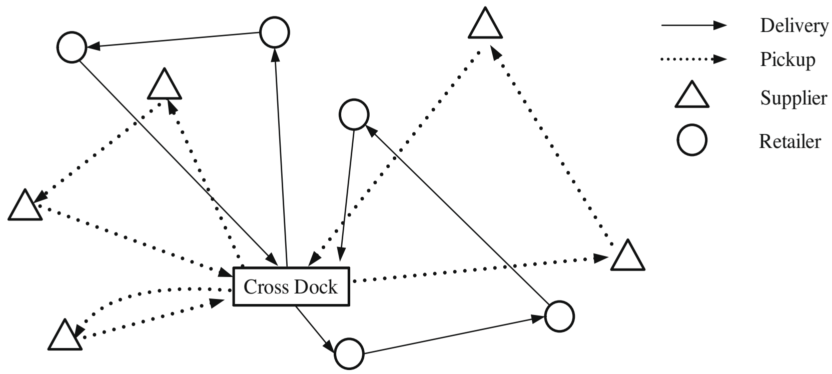

Fig. 2. A proposed network for a single cross-dock.

capacity of vehicles. Detailed steps of the algorithm are summarized as follows.

3.1. Initial solution scheme

1. Pickup process.
   1.1 Calculate the minimum number of vehicles $\underline{v} = \lceil \sum p_i/Q \rceil$.
   1.2 Sequence vehicles in descending order of the remaining space. Let S_j, $j = 1, 2, \ldots, \underline{v}$, be the jth vehicle in the sequence.
   1.3 If all the nodes are assigned, go to Step 1.8; otherwise, continue.
   1.4 Let i be the unassigned node nearest to vehicle S_j.
   1.5 If the completion time of vehicle S_j is smaller than or equal to r and there is remaining capacity, assign node i to vehicle S_j and go to Step 1.2. Otherwise, set $j = j + 1$.
   1.6 If $j < \underline{v} + 1$, go to Step 1.4; otherwise, continue.
   1.7 Set $\underline{v} = \underline{v} + 1$ and go to Step 1.2.
   1.8 Stop.
2. Delivery process.
   2.1 Let r equal T minus the completion time of the pickup process.
   2.2 Repeat the same procedure as in the pickup process to assign retailers to available vehicles.

As shown above, the initial solution scheme is composed of two steps. Step 1 arranges suppliers to vehicles while Step 2 assigns retailers to available vehicles. In Step 1.1, the minimum number of vehicles is calculated by the total pickup quantity divided by the vehicle capacity. Steps 1.2–1.4 are self-explanatory. The completion time in Step 1.5 represents the time that vehicle j starts from the cross-dock and ends at it. Step 1.5 is used to determine whether the nearest node i should be assigned to the jth vehicle. If assigned, then the jth vehicle will stop at node i; otherwise, the next vehicle is considered in Step 1.6. If no vehicle is successfully assigned, then an empty vehicle is added, as expressed in Step 1.7. Steps 1.2–1.7 are repeated until all the nodes are assigned. After completing the pickup process, the delivery process is continued in Step 2, where the time limitation, r, is updated and the same procedure as in the pickup process is repeated to assign retailers to available vehicles.

After obtaining an initial solution, a TS algorithm is developed to further fine-tune the solution. The basic idea of the proposed TS algorithm is to achieve the minimum total cost by continually arranging the nodes from a vehicle to another. There are two major differences between the proposed TS algorithm and Lee et al.'s TS algorithm (2006). First, we arrange a single node to another vehicle one at a time, whereas Lee et al. try to exchange nodes between two vehicles. Second, it is allowed to remove an empty vehicle in the proposed TS algorithm while it is disallowed in Lee et al.'s TS. The detailed steps of the proposed TS algorithm are given as follows, where the tabu size is set to three.

3.2. Proposed TS algorithm

1. Generate an initial solution by using the initial solution scheme described previously.
   Let r equal T minus the completion time of the pickup fleet.
   Let $z = 0$.
2. If $z > 200$, then go to Step 10; otherwise continue.
   If the delivery fleet contains only one vehicle, let $x = 0$, $y = 0$, and go to Step 3; otherwise go to Step 4.
3. Repeat the following steps until $y > 20$.
   3.1. If $x > 20$, go to Step 3.3.
   Randomly select two positions, l_1 and l_2, in the vehicle tour.
   Exchange the nodes in l_1 and l_2.
   3.2. If the new completion time is larger than or equal to r, set $x = x + 1$ and go to Step 3.1.
   Increase each entry f_{ij} by 1 in the frequency matrix whenever node i is assigned to position j.
   If the new transportation cost is smaller, then update the incumbent solution, add l_1 to the tabu list, and set $x = x + 1$.
   Go to Step 3.1.
   3.3. Select the maximum f_{ij} from the frequency matrix and assign node i to position j. Delete row i and column j and select the remaining maximum f_{ij}. Continue the procedure until all nodes are assigned.
   Set the new vehicle tour as the vehicle tour in Step 3.1. Set $x = 0$ and $y = y + 1$.
4. If the delivery fleet contains only one vehicle, then set $z = z + 1$ and go to Step 2.
   Randomly select two vehicles, k_1 and k_2, in the delivery fleet.
   Randomly select a node i from k_1.
5. If there is enough space in k_2, assign node i to the position behind the last node in k_2 and delete an empty vehicle; otherwise, set $z = z + 1$, restore the delivery fleet, and go to Step 2.
6. Let $\bar{v}$ be the number of vehicles in the delivery fleet.
   Let S_j, $j = 1, 2, \ldots, \bar{v}$, be the jth vehicle in the delivery fleet.
   Repeat the following steps until $S_j > \bar{v} + 1$.
   6.1. If the completion time of S_j is smaller than or equal to r, go to Step 6.4; otherwise let $x = 0$ and continue.
   6.2. If $x = 70$, go to Step 6.4.
   Randomly select two positions, l_1 and l_2, in S_j.
   Exchange the nodes in l_1 and l_2.
   6.3. If the completion of S_j is greater than r, set $x = x + 1$; otherwise update the incumbent solution (S_j), add l_1 to the tabu list, and set $x = 0$.
   Go to Step 6.3.
   6.4. Set $j = j + 1$.

7. If the completion time of a vehicle in the delivery fleet is greater than or equal to r, restore the delivery fleet as in Step 4, set $z = z + 1$, and go to Step 4.
8. Reduce the total cost of delivery fleet using the same procedure as in Step 3.
9. If the total cost of the delivery fleet is smaller, update the incumbent solution. Otherwise, restore the delivery fleet as in Step 4 and set $z = z + 1$.
   Go to Step 2.
10. Let r equal T minus the completion time of the delivery fleet and set $z = 0$.
11. Reduce the total cost of pickup fleet using the same procedure as in Steps 2–9.

The proposed TS algorithm is further elaborated in the following paragraphs. First, z and x represent the numbers of iterations of exchanging nodes in a fleet and in a tour, respectively, while y is the number of iterations of generating a new tour. Step 1 generates a feasible solution while updating r and setting $z = 0$. The parameter r is used as the time limitation and z as the number of iterations in the TS. Step 3 is then executed to reduce the transportation cost of a vehicle. Two tasks are performed in Step 3.2: the first task is to record a vehicle tour in the frequency matrix, which is an $n \times n$ matrix for a vehicle tour with n nodes; the second task, to judge whether or not to update the incumbent solution and the tabu list. Step 3.3 describes how to generate a new vehicle tour from the frequency matrix (Ben-Daya & Al-Fawzan, 1998). Steps 4 and 5 carry out the tasks of arranging a node to the other vehicle. If the new arrangement leads to a completion time larger than r, we shorten it in Step 6. In Step 6.3, two positions are randomly selected in a given vehicle tour and the nodes are exchanged in the two positions. If the new completion time is smaller, the incumbent solution and the tabu list are updated, and x is reset. In Step 7, if the completion time of a vehicle is greater than r, the delivery fleet is restored as in Step 4; otherwise, the total cost is reduced in Step 8. Step 9 follows to determine whether the new arrangement should be accepted or not. Steps 2–9 are repeated until z is greater than 200. After finishing the delivery fleet, the same procedure is applied to reduce the total cost of pickup fleet in Step 11.

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
o_k	1000	1000	1000
t_{ij}	U(20,200)	U(20,200)	U(20,200)
c_{ij}	U(48,560)	U(48,480)	U(48,560)
p_i, d_i	U(5,50)	U(5,20)	U(5,30)
No. of suppliers	4	7	12
No. of retailers	6	23	38

4. Computational experiments

In this section, the performance of the proposed TS algorithm is evaluated in comparison to the TS of Lee et al. (2006). To have a fair comparison, both TS algorithms were coded in Java and executed on a Pentium IV Intel processor at 3.2 GHz under Windows XP with 512 MB RAM.

Following Lee et al. (2006), the parameter values are given in Table 1, where the planning horizon, T, was assumed to be 960 min for all the problems and the operational cost of each vehicle was 1000. The numbers of available vehicles for the three problems were 10, 20, and 30, respectively. The transportation cost and the transportation time between two suppliers or two retailers

Table 2
Total cost comparison of initial solution scheme and TS algorithm.

Inst.	Initial solution			TS			PI		
	10	30	50	10	30	50	10	30	50
1	8436.0	11726.0	22964.0	6847.6	7692.9	20704.6	18.8	34.4	9.8
2	9540.0	11752.0	25946.0	6816.8	7787.2	20816.8	28.5	33.7	19.8
3	12114.0	12147.0	22496.0	9615.6	7893.6	19612.2	20.6	35.0	12.8
4	9024.0	11481.0	23967.0	7289.7	7792.2	19549.0	19.2	32.1	18.4
5	9827.0	11811.0	24330.0	6599.0	7224.8	20448.0	32.8	38.8	16.0
6	11105.0	11176.0	24458.0	9324.6	7245.9	21212.0	16.0	35.2	13.3
7	13357.0	12125.0	25996.0	12083.0	8206.9	20640.2	9.5	32.3	20.6
8	9498.0	11374.0	23936.0	8719.6	7880.9	20664.1	8.2	30.7	13.7
9	8947.0	10957.0	23638.0	7362.2	8157.3	18920.0	17.7	25.6	20.0
10	7731.0	12808.0	24204.0	6204.5	7924.7	20384.2	19.7	38.1	15.8
11	8722.0	11441.0	23623.0	7635.3	7452.6	19941.6	12.5	34.9	15.6
12	9770.0	10867.0	20581.0	7867.2	8320.0	17258.4	19.5	23.4	16.1
13	8384.0	12024.0	20857.0	7097.9	8222.7	17829.9	15.3	31.6	14.5
14	5546.0	11710.0	24615.0	5208.0	8211.7	19845.2	6.1	29.9	19.4
15	8085.0	12172.0	25405.0	7103.2	8144.6	21863.0	12.1	33.1	13.9
16	9689.0	10268.0	22843.0	8768.7	7451.7	20144.2	9.5	27.4	11.8
17	9424.0	13422.0	24639.0	9003.0	8086.2	20093.3	4.5	39.8	18.4
18	8393.0	11546.0	24129.0	6887.5	7576.0	20244.8	17.9	34.4	16.1
19	8557.0	11758.0	22777.0	7123.0	7871.2	19955.0	16.8	33.1	12.4
20	12586.0	11898.0	22363.0	10471.0	7883.7	19267.7	16.8	33.7	13.8
21	7258.0	12515.0	23879.0	5431.4	7914.1	19533.4	25.2	36.8	18.2
22	8796.0	12916.0	23667.0	6908.0	8005.3	19032.1	21.5	38.0	19.6
23	10212.0	10973.0	24663.0	9224.1	7883.5	20562.5	9.7	28.2	16.6
24	11976.0	11507.0	19877.0	11976.0	7731.2	19288.2	0.0	32.8	3.0
25	8899.0	11514.0	21819.0	6638.0	7884.8	19695.9	25.4	31.5	9.7
26	9943.0	12458.0	22176.0	7216.9	8001.6	20610.5	27.4	35.8	7.1
27	11749.0	12641.0	22679.0	9709.8	8899.4	18942.8	17.4	29.6	16.5
28	9267.0	10131.0	22814.0	7408.0	10131.0	20097.3	20.1	0.0	11.9
29	7583.0	12952.0	24034.0	6748.5	8276.9	22248.1	11.0	36.1	7.4
30	9170.0	11433.0	24007.0	7304.4	8251.6	19321.9	20.3	27.8	19.5
Avg.	9452.9	11783.4	23446.1	7886.4	8000.2	19957.6	16.7	31.8	14.7

Table 3
Computation time comparison of initial solution scheme and TS algorithm.

Inst.	Initial solution			TS		
	10	30	50	10	30	50
1	0.00	0.00	0.00	0.22	0.37	0.49
2	0.00	0.00	0.00	0.23	0.16	0.64
3	0.00	0.00	0.00	0.19	0.43	0.30
4	0.00	0.00	0.00	0.27	0.23	0.44
5	0.00	0.00	0.00	0.21	0.39	0.61
6	0.00	0.00	0.00	0.03	0.22	0.56
7	0.00	0.00	0.00	0.01	0.11	0.55
8	0.00	0.00	0.00	0.04	0.16	0.42
9	0.00	0.00	0.00	0.25	0.16	0.38
10	0.00	0.00	0.00	0.36	0.20	0.52
11	0.00	0.00	0.00	0.00	0.29	0.37
12	0.00	0.00	0.00	0.24	0.16	0.17
13	0.00	0.00	0.00	0.24	0.15	0.16
14	0.00	0.00	0.00	0.00	0.18	0.53
15	0.00	0.00	0.00	0.41	0.38	0.55
16	0.00	0.00	0.00	0.03	0.28	0.30
17	0.00	0.00	0.00	0.06	0.34	0.44
18	0.00	0.00	0.00	0.19	0.28	0.53
19	0.00	0.00	0.00	0.03	0.36	0.28
20	0.00	0.00	0.00	0.00	0.24	0.36
21	0.00	0.00	0.00	0.00	0.26	0.61
22	0.00	0.00	0.00	0.01	0.35	0.37
23	0.00	0.00	0.00	0.11	0.38	0.51
24	0.00	0.00	0.00	0.00	0.48	0.05
25	0.00	0.00	0.00	0.10	0.20	0.33
26	0.00	0.00	0.00	0.03	0.16	0.14
27	0.00	0.00	0.00	0.06	0.17	0.31
28	0.00	0.00	0.00	0.01	0.00	0.39
29	0.00	0.00	0.00	0.18	0.38	0.20
30	0.00	0.00	0.00	0.09	0.28	0.65
Avg.	0.00	0.00	0.00	0.12	0.26	0.41

were generated asymmetrically. Loading and unloading quantities were generated by discrete uniform distributions.

Before conducting a formal experiment, it is necessary to determine appropriate parameter values for the proposed algorithm. To determine the values of r, it was examined starting from 380 to 600 min with an increment of 10 min. According to the testing results, the best parameter values of r were 390, 560, 410 for the three problems, respectively. Percentage improvement (PI) is then used as the performance measure and is computed according to the following equation:

$$PI = 100 \times \frac{\text{total cost of Lee et al.'s TS} - \text{total cost of the proposed TS}}{\text{total cost of Lee et al.'s TS}}$$

The first experiment was performed to evaluate the proposed TS algorithm along with the embedded initial solution scheme, where 30 instances were run for each of the three problems. For each instance, 10 replications were made and the average was recorded. Tables 2 and 3 summarize the solutions and the computation times for both the initial solution scheme and the TS algorithm. With noticeable outcomes, the initial solution scheme can find a good feasible solution in a very short time, and the TS algorithm can further perfect the solution with an average improvement of 16.7%, 31.8% and 14.7% for the three problems, respectively. The main reason why the proposed TS performed better than the Lee et al.'s TS is that the former continues trying to reduce the number of required vehicles while the number of vehicles is fixed once determined by the initial solution scheme in the latter approach.

There followed the second experiment for the purpose of comparing the proposed TS with Lee et al.'s TS (2006) through 30 instances for each problem. The value of each instance was reported from the average of 10 repetitions. Comparative results of solutions and computation times are summarized in Tables 4 and 5, which indicate that the proposed TS algorithm yields superior results while using much less computation time. The average

Table 4
Total cost comparison of proposed TS and Lee et al.'s TS.

Inst.	Lee et al.'s TS			Proposed TS			PI		
	10	30	50	10	30	50	10	30	50
1	7571.4	12366.7	24284.6	6847.6	7692.9	20704.6	9.6	37.8	14.7
2	7103.7	14173.0	23435.6	6816.8	7787.2	20816.8	4.0	45.1	11.2
3	9993.5	13836.8	23449.4	9615.6	7893.6	19612.2	3.8	43.0	16.4
4	8338.0	10995.4	23471.1	7289.7	7792.2	19549.0	12.6	29.1	16.7
5	8709.9	11757.8	23406.2	6599.0	7224.8	20448.0	24.2	38.6	12.6
6	9143.5	11027.7	24026.6	9324.6	7245.9	21212.0	−2.0	34.3	11.7
7	12721.2	11899.2	24190.0	12083.0	8206.9	20640.2	5.0	31.0	14.7
8	9275.7	12825.5	23158.9	8719.6	7880.9	20664.1	6.0	38.6	10.8
9	8096.5	12718.6	23594.7	7362.2	8157.3	18920.0	9.1	35.9	19.8
10	7044.8	11794.7	23530.5	6204.5	7924.7	20384.2	11.9	32.8	13.4
11	8051.8	12094.9	23371.7	7635.3	7452.6	19941.6	5.2	38.4	14.7
12	8661.0	12132.5	21082.8	7867.2	8320.0	17258.4	9.2	31.4	18.1
13	7370.2	13223.4	21610.7	7097.9	8222.7	17829.9	3.7	37.8	17.5
14	7132.3	12413.9	23397.9	5208.0	8211.7	19845.2	27.0	33.9	15.2
15	7563.4	12521.4	24041.9	7103.2	8144.6	21863.0	6.1	35.0	9.1
16	9983.6	12044.4	22893.4	8768.7	7451.7	20144.2	12.2	38.1	12.0
17	9538.1	12699.4	22950.4	9003.0	8086.2	20093.3	5.6	36.3	12.4
18	8057.4	11001.4	24358.2	6887.5	7576.0	20244.8	14.5	31.1	16.9
19	9042.6	12724.4	25068.7	7123.0	7871.2	19955.0	21.2	38.1	20.4
20	10478.0	12357.7	23232.1	10471.0	7883.7	19267.7	0.1	36.2	17.1
21	8380.5	13177.0	22564.8	5431.4	7914.1	19533.4	35.2	39.9	13.4
22	9016.9	11545.0	24360.7	6908.0	8005.3	19032.1	23.4	30.7	21.9
23	9489.2	12308.1	24377.8	9224.1	7883.5	20562.5	2.8	35.9	15.7
24	12513.6	12722.7	22008.7	11976.0	7731.2	19288.2	4.3	39.2	12.4
25	7114.3	12844.9	24256.6	6638.0	7884.8	19695.9	6.7	38.6	18.8
26	8421.3	13297.5	23424.9	7216.9	8001.6	20610.5	14.3	39.8	12.0
27	10666.8	13415.2	22961.4	9709.8	8899.4	18942.8	9.0	33.7	17.5
28	10123.3	12613.0	23822.3	7408.0	10131.0	20097.3	26.8	19.7	15.6
29	7503.2	12840.8	23678.3	6748.5	8276.9	22248.1	10.1	35.5	6.0
30	7642.6	13796.2	23149.8	7304.4	8251.6	19321.9	4.4	40.2	16.5
Avg.	8824.9	12505.6	23438.7	7886.4	8000.2	19957.6	10.6	36.0	14.9

Table 5
Computation time comparison of proposed TS and Lee et al.'s TS.

Inst.	Lee et al.'s TS			Proposed TS			PI		
	10	30	50	10	30	50	10	30	50
1	1.52	3.00	5.69	0.22	0.37	0.49	85.5	87.7	91.4
2	1.74	3.55	5.73	0.23	0.16	0.64	86.8	95.5	88.8
3	2.37	4.32	5.80	0.19	0.43	0.30	92.0	90.0	94.8
4	1.60	2.09	5.87	0.27	0.23	0.44	83.1	89.0	92.5
5	2.28	2.26	7.28	0.21	0.39	0.61	90.8	82.7	91.6
6	1.82	2.10	5.38	0.03	0.22	0.56	98.4	89.5	89.6
7	2.80	2.61	7.65	0.01	0.11	0.55	99.6	95.8	92.8
8	1.85	3.00	9.88	0.04	0.16	0.42	97.8	94.7	95.7
9	2.04	3.10	5.54	0.25	0.16	0.38	87.7	94.8	93.1
10	1.82	2.38	5.77	0.36	0.20	0.52	80.2	91.6	91.0
11	1.80	3.03	5.37	0.00	0.29	0.37	100.0	90.4	93.1
12	1.72	2.64	4.46	0.24	0.16	0.17	86.0	93.9	96.2
13	1.54	2.92	4.62	0.24	0.15	0.16	84.4	94.9	96.5
14	1.53	2.76	5.44	0.00	0.18	0.53	100.0	93.5	90.3
15	1.61	3.06	6.31	0.41	0.38	0.55	74.5	87.6	91.3
16	2.05	3.15	5.18	0.03	0.28	0.30	98.5	91.1	94.2
17	2.29	2.14	6.93	0.06	0.34	0.44	97.4	84.1	93.7
18	1.74	1.70	6.25	0.19	0.28	0.53	89.1	83.5	91.5
19	2.21	2.77	5.68	0.03	0.36	0.28	98.6	87.0	95.1
20	2.55	2.72	4.79	0.00	0.24	0.36	100.0	91.2	92.5
21	2.06	3.39	5.43	0.00	0.26	0.61	100.0	92.3	88.8
22	2.42	2.43	6.04	0.01	0.35	0.37	99.6	85.6	93.9
23	2.31	2.93	5.88	0.11	0.38	0.51	95.2	87.0	91.3
24	2.64	2.87	5.36	0.00	0.48	0.05	100.0	83.3	99.1
25	1.68	2.67	5.76	0.10	0.20	0.33	94.0	92.5	94.3
26	2.04	3.31	5.05	0.03	0.16	0.14	98.5	95.2	97.2
27	2.47	3.25	5.17	0.06	0.17	0.31	97.6	94.8	94.0
28	2.69	2.60	5.56	0.01	0.00	0.39	99.6	100.0	93.0
29	1.73	3.28	64.84	0.18	0.38	0.20	89.6	88.4	99.7
30	1.83	3.78	5.91	0.09	0.28	0.65	95.1	92.6	89.0
Avg.	2.02	2.86	7.82	0.12	0.26	0.41	94.1	90.9	94.8

improvements are 10.6%, 36.0% and 14.9% for the three problems, respectively. As for the computation time, Lee et al.'s TS required 2.02, 2.86 and 7.82 s and the proposed TS took only 0.12, 0.26 and 0.41 s for the three problems, respectively. As a result, we can thus conclude that our TS algorithm performs better than Lee et al.'s TS algorithm.

5. Conclusions

Although cross-docking has been widely practiced within both manufacturing and retailing companies and brings benefits to companies, there are very few studies on the integration of vehicle routing problem and cross-docking. Lee et al. (2006) are the first who proposed an integration model for the problem. Their strategy is to consolidate all goods received from the suppliers at the cross-dock and dispatch them to the customers, which has stimulated a need to expand the algorithm by considering both the pickup and delivery processes simultaneously. In this study, we have proposed a new TS algorithm which employs a completely different approach from Lee et al.'s TS. Computational results show that the proposed TS algorithm provides a superior solution while using much less computation time. The average improvements are as high as 10–36% for different size of problems. The main reason why the proposed TS performed better than the Lee et al.'s TS is that the former continues trying to reduce the number of required vehicles while the number is fixed once determined by the initial solution scheme in the latter approach.

As one of the future research activities, the proposed TS algorithm can be further extended to the cases where receiving time limitations are considered for each supplier and/or retailer. Also, for practical use in real-world business applications, another extended research is the development of a model which allows split deliveries. If deliveries can be split, we can achieve a higher utilization of vehicles. Further another research project involves a combination of vehicle routing problem and cross-docking in a reverse logistic system for the reuse of products and materials (Dobos, 2003).

References

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

Barbarosoglu, G., & Ozgur, D. (1999). A tabu search algorithm for the vehicle routing problem. *Computers and Operations Research, 26*, 255–270.

Ben-Daya, M., & Al-Fawzan, M. (1998). A tabu search approach for the flow shop scheduling problem. *European Journal of Operational Research, 109*, 88–95.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society, 53*, 512–522.

Dobos, I. (2003). Optimal production-inventory strategies for HMMS-type reverse logistics system. *International Journal of Production Economics, 81–82*, 351–360.

Gumus, M., & Bookbinder, J. H. (2004). Cross-docking and its implications in location-distribution systems. *Journal of Business Logistics, 25*, 199–299.

Kreng, V. B., & Chen, F. T. (2008). The benefits of a cross-docking delivery strategy: A supply chain collaboration approach. *Production Planning and Control, 19*, 229–241.

Lee, Y. H., Jung, W. J., & Lee, K. M. (2006). Vehicle routing scheduling for cross-docking in the supply chain. *Computer and Industrial Engineering, 51*, 247–256.

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

Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2003). *Designing and managing the supply chain*. New York: McGraw-Hill.

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.

Tzoreff, T. E., Granot, D., Granot, F., & Sosic, G. (2002). The vehicle routing problem with pickups and deliveries on some special graphs. *Discrete Applied Mathematics, 119*, 193–229.

Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., & Laporte, G. (2008). Vehicle routing with cross-docking. *Journal of the Operational Research Society*. doi:10.1057/jors.2008.108 [advance online publication].