International Journal of Management Science
Vol 3, No 1, May 1997

A HEURISTIC ALGORITHM FOR VEHICLE ROUTING PROBLEM WITH BACKHAULS

NAE-HEON KIM · SUK-CHUL RIM · BYUNG-DO MIN
Department of Industrial Engineering, Ajou University
Suwon 442-749, Korea

ABSTRACT

We address the problem of determining the sequence of a vehicle with fixed capacity to visit n nodes at which a predetermined amount is picked up and/or delivered. The objective is to minimize the total travel distance of the vehicle, while satisfying the pick-up/delivery requirements and feasibility at all nodes. Existing methods for the problem allows the vehicle to visit a node twice, which is impractical in many real situations. We propose a heuristic algorithm, in which every node is visited exactly once. Computational results using random problems indicate that the proposed heuristic outperforms existing methods for practical range of the number of nodes in reasonable computation time.

1. INTRODUCTION

The Vehicle Routing Problem(VRP) deals with a set of delivery customers, with known demands, to be serviced by a homogeneous set of vehicles of fixed capacity from a single distribution center(DC). The objective of the VRP is to design a set of routes such that the total distance traveled by all vehicles is minimized, while all delivery customers are serviced; and the demands of the customers assigned to each route will not exceed the vehicle capacity[3].

The Vehicle Routing Problem with Backhauls(VRPB) is a variation of the VRP, considering both delivery and pick-up customers. Delivery customers are the nodes which need to receive a certain amount of goods from the single DC, while pick-up customers are the nodes which need to send a certain amount of goods back to the DC by the same vehicle.

There are many practical applications of VRPB in which customers need both delivery and pick-up services; see Bodin et al.[2], Fresh Air Fund annual report[5], or Golden and Assad [7]. Additional practical examples for the VRPB include the followings: beverage industry needs to consider not only the delivery of bottled products but also pick-up of empty bottles, because of both regulations and economic reason. Such pick-up problems also occur in the gas industry where expensive metal cylinders are used for specific gases such as oxygen or hydrogen for industrial processes such as semiconductor manufacturing. Collecting pallets in distribution industry is a similar problem, although the volume of delivery is different from that of pick-up.

Although Anily[1], Dief and Bodin[4], Goetschalckx and Jacobs-Blecha[6], and Yano et al.[11] addressed similar problems, those studies deal with the problem in which all deliveries must be made before pick-up starts. However, in this paper we address a relaxation of the problem in that the vehicle is allowed to visit the nodes in any sequence so as to minimize the total travel distance, as long as the feasibility is satisfied at all nodes.

For such a relaxed problem, the work by Mosheiov[10] is the only one reported in the literature. The author presented a mixed integer linear programming formulation, and solved the problems of up to 12 nodes. He also proposed two heuristic algorithms, namely, PDαT and Cheapest Feasible Insertion(CFI). The basic idea of PDαT is that, for any tour that visits all nodes exactly once, there exists a node S such that starting from S to follow the tour gives a feasible solution for the VRPB. If the initial tour for n nodes (excluding the origin) is obtained by using an α-optimal algorithm (that is, the solution is guaranteed not to exceed the optimal value by a factor of α), then the algorithm is called PDαT.

The CFI heuristic, on the other hand, is based on inserting pick-up points one by one into a basic delivery tour (obtained by an exact or heuristic algorithm) such that the incremental distance will be minimized, while maintaining the feasibility. In CFI, the customers having both pick-up and delivery demands are treated as two different customers located at the same location.

The motivation to develop an alternative heuristic algorithm for the problem is based on the potential weaknesses of Mosheiov's two heuristics. More specifically, PDαT will perform well for the problems in which the DC is located at the center. However, it will very likely perform poorly for the problems in which the DC is located at the corner, which will be the case if the DC operates multiple vehicles, and each vehicle covers a section. For CFI, due to the myopic behavior of the algorithm, the vehicle may visit a node Q for delivery, and visit other nodes before it returns to the node Q for pick-up. In practical sense, whenever a vehicle visits a node for either pick-up or delivery, considerable setup time and effort is required; or simply the customer at the node may want to perform both delivery and pick-up at once. For such reasons, allowing the vehicle to visit a node twice will probably be impractical.

Hence, in this paper we will propose an algorithm which forces each customer with nonzero delivery or pick-up demand to be visited exactly once, just as in the Traveling Salesman Problem(TSP).
The paper is organized as follows: we will present a mathematical formulation of the problem in section 2; explain the heuristic algorithm we propose in section 3; report the computational results in section 4 that the proposed algorithm outperforms Mosheiov's algorithm for practical range of the number of nodes in reasonable computation time; and suggest some future research issues in section 5.

2. PROBLEM FORMULATION

In the VRPB, determining the sequence of visits may be affected by the priority between delivery and pick-up, or the value of goods to be picked up. In this study, however, we will only consider the problem in which single vehicle is already assigned to a set of customers; and delivery after pick-up causes no additional cost. We will use the following notations:

- V : set of all customers, not including the DC, where $|V| = n-1$,
- c_{ij} : distance between customer i and j, $1 \le i,j \le n$,
- d_i : delivery demand to location i, $2 \le i \le n$,
- p_i : pick-up demand at location i, $2 \le i \le n$,
- K : vehicle capacity
- x_{ij} =1 if arc(i,j) is used by the vehicle; 0 otherwise,
- y_i : total amount picked up by the vehicle up to customer i including p_i, and
- z_i : total amount to be delivered to the customers after customer i.

The problem can be described as follows: the vehicle of capacity K starts at DC; visits n-1 customers exactly once for pick-up and/or delivery; and returns to DC. It is assumed that d_i and p_i are known constants; and both the total delivery and pick-up quantities are equal to the vehicle capacity; that is, $\sum d_i = \sum p_i = K$. When any one of these two quantities exceeds the capacity of the vehicle, there is no feasible solution. On the other hand, when the total delivery (pick-up) is less than the capacity K, the problem can be converted to the above "standard form" by adding a dummy delivery (pick-up) customer with demand of $K - \sum d_i$ ($\sum p_i$) located at the DC. (Readers may refer to the beginning of section 2 of Mosheiov[10] for further explanation.) The vehicle has to visit each customer with nonzero demand exactly once; without exceeding the vehicle capacity. The objective is to minimize the total travel distance of the vehicle. We formulate the problem as follows:

$$\text{Min} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \ldots, n \qquad \ldots\ldots(1)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots n \qquad \ldots\ldots(2)$$

$$\sum_{j=1}^{n} y_i x_{ij} - \sum_{k=1}^{n} y_k x_{ki} = p_i, \quad i = 2, \ldots, n \qquad \ldots\ldots(3)$$

$$\sum_{j=1}^{n} z_i x_{ij} - \sum_{k=1}^{n} z_k x_{ki} = -d_i, \quad i = 2, \ldots, n \qquad \ldots\ldots(4)$$

$$\sum_{k=2}^{n} y_k x_{k1} = K \qquad \ldots\ldots(5)$$

$$\sum_{k=2}^{n} z_k x_{k1} = 0 \qquad \ldots\ldots(6)$$

$$y_1 = 0 \qquad \ldots\ldots(7)$$

$$z_1 = K \qquad \ldots\ldots(8)$$

$$y_i + z_i \leq K, \quad i = 1, \ldots, n \qquad \ldots\ldots(9)$$

$$\sum_{j=1}^{n} (y_i + z_i) x_{ij} - d_j + p_j \leq K, \quad i = 1, \ldots, n \qquad \ldots\ldots(10)$$

$$x_{ij} = 0 \text{ or } 1, \quad i, j = 1, \ldots, n$$

$$\text{and } y_i \geq 0, \; z_i \geq 0, \quad i = 1, \ldots, n \qquad \ldots\ldots(11)$$

Constraints (1) and (2) guarantee that each customer is visited exactly once by the vehicle; (3) and (4) indicate that, as the vehicle visits node i, the total pick-up(delivery) amount will be increased(decreased) by $p_i(d_i)$; (6) and (8) imply that the vehicle starts its tour from DC fully loaded, and delivers all the delivery loads to the customers; (5) and (7) indicate that the vehicle returns to DC fully loaded with pick-up loads; (9) forces the sum of pick-up load and the remaining delivery load at any customer not to exceed the vehicle capacity throughout the tour; (10) implies that customer j can be reached from customer i only if the additional load at j does not cause the total load after visiting j to exceed the vehicle capacity.

Exact solution procedures for VRP can be applied to only small problems even when pick-up and delivery are ignored. Furthermore, the above formulation contains nonlinear constraints (3)-(6) and (10). Therefore, we will propose a heuristic algorithm for the problem in the following section.

3. HEURISTIC ALGORITHM

A two-phase heuristic is proposed in this section. In the first phase, a feasible tour is constructed. Subsequently, in the second phase, this feasible tour is improved while maintaining the feasibility.

3.1 Initial feasible tour

Consider a set of nodes V which includes n-1 nodes to visit. Then V can be divided into two mutually exclusive and exhaustive subsets V^+ and V^-, where $V^+ = \{i \in V \mid d_i \geq p_i, i=2,..,n\}$; and $V^- = \{i \in V \mid d_i < p_i, i=2,..,n\}$. That is, the nodes in V^+ "generate" the space in the vehicle, while the nodes in V^- "consume" the space. A natural feasible tour for VRPB is to visit all nodes in V^+ in any sequence, and subsequently, visit all nodes in V^- in any sequence. The above algorithm for obtaining an initial feasible tour can be described as follows:

Algorithm for Initial Tour

Step 1. Include DC in V^+ and V^-, to obtain W^+ and W^-, respectively.

Step 2. Using any heuristic method, construct directed Hamiltonian paths T^+ and T^- for W^+ and W^-, respectively, which start from DC, and return to DC.

Step 3. Omit DC from T^- to obtain P^-

Step 4. Insert P^- between DC and the preceding node in T^+ to obtain an initial feasible tour.

We can consider two other directed paths, Q^+ and R^-, whose directions are the opposite of T^+ and P^-, respectively. Combining the two paths(T^+ and Q^+) for V^+; and the two paths(P^- and R^-) for V^- in step 2 through 4 described above, we can obtain four feasible tours. In the following, we will obtain four solutions by using these four initial solutions in order to select the best one. Depending on how the initial tours are constructed, the solution quality may differ. In section 4, we will compare the solution quality in which Q^+, R^-, T^+ and P^- are obtained by optimal solution procedure, nearest neighbor heuristic, and random sequencing, respectively.

3.2 Improvement of initial solutions

The initial feasible solutions obtained in the previous section can be improved by applying the next two improvement procedures.

(1) Best Feasible Insertion (BFI) Procedure

Consider a feasible tour U which comprises two subtours U^+ and U^- obtained from section 3.1 for the nodes in V^+ and V^-, respectively. The nodes in U^- can be moved to the front in U^+ as long as the tour remains feasible. The following algorithm describes the procedure to select the "best" node among the nodes initially in U^-, and move to the "best" position in the subtour U^+, one by one, in order to maximally improve the solution quality, while maintaining the feasibility.

<u>Algorithm BFI</u>

Step 0. Set $S=V^-$, $T=W^+$.

Step 1. If $S=\emptyset$, go to 4; otherwise, for every node $i\in S$, find all feasible insertion positions in U^+. If no such position is found, go to step 4; otherwise go to step 2.

Step 2. Compute savings for all possible insertions. If all savings are non-positive, go to step 4.

Step 3. Find the node $k\in S$ with the largest positive saving. (In case of tie, break tie arbitrarily.) Insert k into U^+ at the position such that the largest saving can be achieved. Connect the preceding node and succeeding node of k in U^-. Remove k from U^-. Update $T=T\cup\{K\}$; $S=S-\{K\}$. and Go to Step 1.

Step 4. Obtain the tour by attaching U^- to the end of U^+; stop.

Note that BFI is different from Mosheiov's CFI algorithm as follows: consider a VRPB with n nodes, each of which has both delivery and pick-up demands. Recall that, in CFI, if a node has both delivery and pick-up demands, then the node is considered as the two different nodes with identical location, that is, CFI begins with $2n$ nodes (n "delivery" nodes and n "pick-up" nodes). As each "pick-up" node is "inserted" to the subtour comprising the "delivery" nodes, if a node is inserted adjacent to its "pair" node, the vehicle will visit the node only once; otherwise the vehicle will visit the same node twice. On the other hand, since BFI of the proposed algorithm does not split the node into a "delivery" node and a "pick-up" node, each node is guaranteed to be visited exactly once. The myopic selection of the insertion location in CFI may result in a number of nodes visited twice. We expect that the vehicle travel distance can be reduced by forcing each node visited exactly once using BFI.

The best solution among the four solutions obtained by applying the BFI to the four initial feasible solutions defined in Section 3.1. can be used as a suboptimal solution to the problem. This suboptimal solution can be further improved by exchanging the patterns in it.

(2) Pattern Exchange Procedure(PEP)

A pattern in a tour is defined as follows:

Definition Consider a series of $k+1$ nodes (n_1, ..., n_k, n_{k+1}) not including DC in a tour. ξ = (n_1, ..., n_k) is called a pattern of size k ($k \geq 1$) if ξ = (n_1, ..., n_k) appears in all four solutions obtained by BFI; but (n_1, ..., n_k, n_{k+1}) does not..

The nodes in a pattern can be treated as a single node whose delivery (pick-up) demand is the sum of the delivery (pick-up) demands of each node in the pattern. The distance from pattern A to pattern B is defined as the distance from the last node of the pattern A to the first node of the pattern B. The pattern exchange procedure is defined as follows:

Step 0. Identify all patterns from the four solutions obtained from BFI by using the definition of pattern. Regard each pattern as a single node to obtain the reduced problem.
Step 1. For each pattern, compute the total delivery and pick-up demands.
Step 2. Calculate the distance matrix for the reduced problem.
Step 3. Apply the algorithm for initial tour described in section 3.1 to the reduced problem in order to generate the four feasible solutions.
Step 4. Apply the BFI to the four feasible solutions obtained in step 3; and choose the best solution.

Although the BFI and PEP are not new approaches, these can be effectively used for strengthening the proposed algorithm. In the following section, we will evaluate the solution quality of the proposed algorithm.

4. COMPUTATIONAL RESULTS

In order to examine the error bound of the proposed algorithm, we will first compare the solution obtained by the proposed heuristic with the optimum value obtained by complete enumeration for small problems. Subsequently, we will examine the solution quality of the proposed heuristic by comparing with those obtained by Mosheiov's two heuristic algorithms for larger problems with up to 100 nodes.

For small problems with up to eleven nodes, test problems are generated randomly and solved by the proposed heuristic algorithm. In generating the random problems, we consider the following variables:

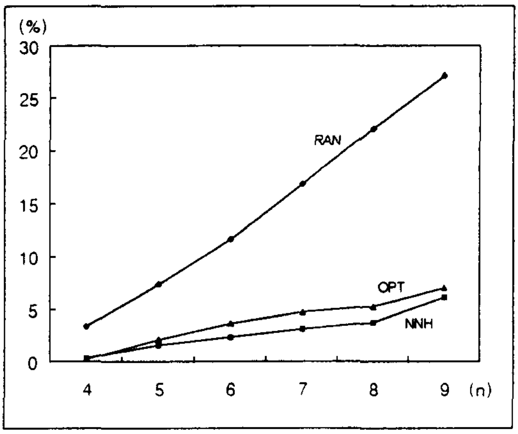
(1) number of nodes: n = 4, 5, 6, 7, 8, 9, 10, 11

(2) location of the origin (or DC): at a corner or at the center of the nodes' location area.

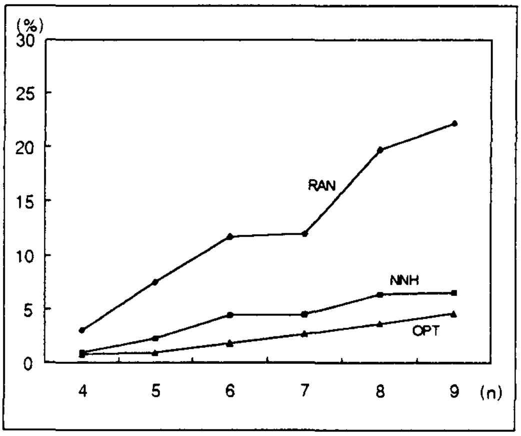
Recall that, in the proposed algorithm, we can determine the sequence of the nodes in V^+ and V^-, by using three methods; namely, random sequence, nearest neighbor heuristic(NNH), and optimal by enumeration. We will first examine the relative performance among the three methods. For various n values and the two possible location of DC, 1000 random problems are generated in order to compare the solution quality. In generating the locations of nodes, both x and y integer coordinates are obtained from a discrete uniform distribution between 1 and 100. Integer values for d_i and p_i are randomly generated such that $\sum d_i = \sum p_i = 10n$, where n is the number of nodes.

Before the computational results are shown, we first define the terms regarding the errors as follows. Let e^L and e^S be the largest and smallest error among the four solutions over the optimal solution, respectively. Then, for the 1,000 tested random problems, the "maximum error" is defined as the largest error among 1,000 e^L's. The "average maximum error" is defined as the average of the 1,000 e^L's. The "average minimum error" is defined as the average of the 1,000 e^S's. In other words, "maximum error" is the worst possible error we can experience when solving 1,000 problems if only one initial feasible solution is considered out of four possible initial solutions for each problem. The "average maximum error" means the average of 1,000 maximum errors out of the four solutions. This is the expected largest error if only one initial feasible solution is used. The "average minimum error" is the expected error when all four initial feasible solutions are used and the best result is selected (this is also called "multiple run"). That is, the difference between the average maximum error and the average minimum error represents the "maximum" contribution of the multiple run.

First, we will examine how much difference will result as we use different methods for determining the two subtours U^+ and U^- in the algorithm BFI. Figure 1 shows the average minimum errors of the three solutions (using multiple run, but without applying pattern exchange) over the optimal solution



(a) when DC is at the corner



(b) when DC is at the center

Figure 1. Average minimum errors

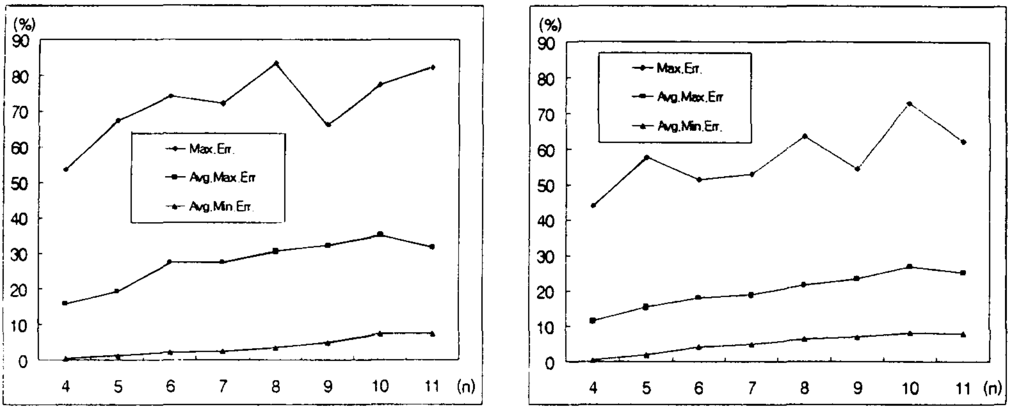
(obtained by complete enumeration). The three solutions of the proposed algorithm denoted by RAN, OPT, and NNH in Figure 1 designate the solution obtained from determining the sequences of nodes in V^+ and V^- randomly, optimally using enumeration, or using nearest neighbor heuristic, respectively.

Figure 1 indicates that the solution using random sequencing of nodes in V^+ and V^- results in more than 25% error over the optimal solution for nine node problems. However, the solutions obtained by using enumeration or NNH show less than 7% error over the optimum for nine node problems, which can be regarded as acceptable heuristic solution. Figure 1 implies that the quality of the subtours for V^+ and V^- affects the solution quality of the final solution of the proposed algorithm.

An interesting observation is that NNH outperforms OPT when DC is located at the corner, as shown in Figure 1(a), while OPT outperforms NNH when DC is located at the center, as shown in Figure 1(b). Note that the solution obtained from NNH performs reasonably well, in much shorter time than enumeration. Furthermore, recall that as multiple vehicles are used, the DC will be located at the corner of the customer locations for each vehicle, in which case NNH outperforms OPT. Hence, we will examine the performance of the solution obtained from only NNH in the following analysis.

Next, we will examine how much additional error will result if we consider only one initial feasible tour rather than all four possible initial feasible tours. Figure 2 compares the three different errors obtained from solving 100 random problems. Recall that four initial feasible solutions are used in BFI in order to obtain the four solutions. Figure 2 shows that multiple run significantly reduces the error in the proposed algorithm in either configuration.

Subsequently, we will examine the impact of pattern exchange on the



(a) when DC is at the corner (b) when DC is at the center

Figure 2. Magnitude of errors

relative error of the solution. Figure 3 shows the impact of the pattern exchange on the NNH solution. The figure implies that approximately 1% of error can be reduced by applying the pattern exchange. The figure also shows that our heuristic performs better when the DC is located at the corner of the nodes than at the center. However, as the number of nodes increases, the impact of the location of DC seems to be decreased.

So far, we have dealt with problems of up to 11 nodes because obtaining the optimal solution by complete enumeration takes prohibitively long computation time for $n \geq 12$. Hence, for larger problems, we will compare the performance of the proposed algorithm with that of Mosheiov's algorithm[10].

One hundred random problems are solved for problems with 8 nodes to 60 nodes; and fifty random problems are solved for problems with 70 nodes to

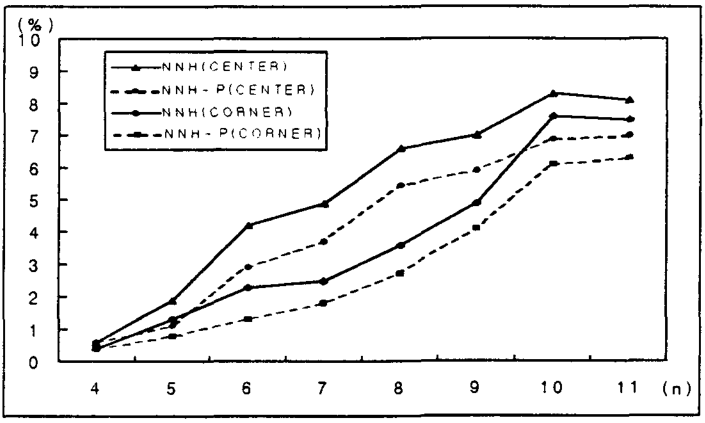


Figure 3. Impact of pattern exchange on the relative error

100 nodes. As mentioned in the previous section, the DC can be located at either center or corner of the customer locations. Here we consider only the cases where DC is located at the corner of the customer locations, because as the DC serves an area with many vehicles, it is more likely that each vehicle will serve a section of the area, in which case a vehicle will start from the corner of the customer locations. For fair comparison with Mosheiov's algorithm, pattern exchange is not performed in the proposed algorithm.

Table 1 summarizes the results of the computation. The average distance of the vehicle obtained by using Mosheiov's two heuristic algorithms (namely, PD αT and CFI) and the proposed algorithm are shown in Figure 4. The average of 100 ir 50 improvements of the proposed algorithm over the two existing algorithms by Mosheiov are shown in Figure 5. (Note that the average improvement does not mean the ratio of the two average distances in Table 1.) Figure 5 indicates that, for practical range of $8 \leq n \leq 30$, the improvement of the proposed algorithm ranges between 10% and 6%, which is significant reduction in daily vehicle routing operations.

For larger problems, the computation time must be examined to see if the algorithms take prohibitively long computation time. It turns out that the proposed algorithm does not require excessively long computational time

Table 1. Comparison of performance for large problems

# nodes	# tested problems	Average Distance			Avg. improvement over	
		Mosheiov's		Proposed	PD αT	CFI
		PD αT	CFI			
8	100	396.11	381.72	338.82	12.68%	9.80%
12		464.36	430.81	392.60	14.02%	8.03%
16		516.97	486.77	441.40	13.09%	8.30%
20		576.93	532.48	486.50	14.66%	7.67%
30		668.54	624.14	584.92	11.53%	5.71%
40		757.89	698.58	669.03	10.98%	3.55%
50		816.79	767.65	737.22	8.91%	3.58%
60		891.06	825.51	804.16	9.08%	2.05%
70	50	955.30	873.14	869.90	8.90%	0.12%
80		997.12	950.76	941.80	5.01%	0.64%
90		998.41	946.22	977.63	5.55%	0.38%
100		1042.14	992.81	1021.56	5.51%	0.72%

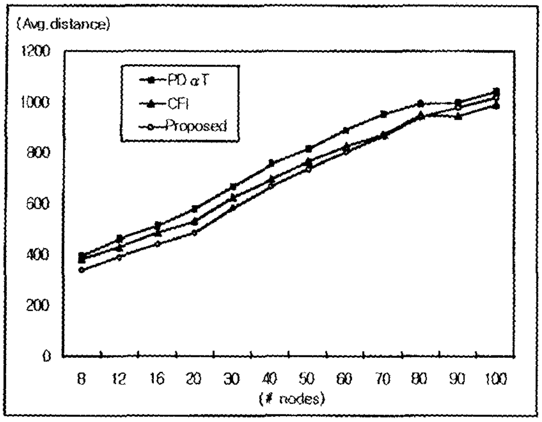


Figure 4. Average distance

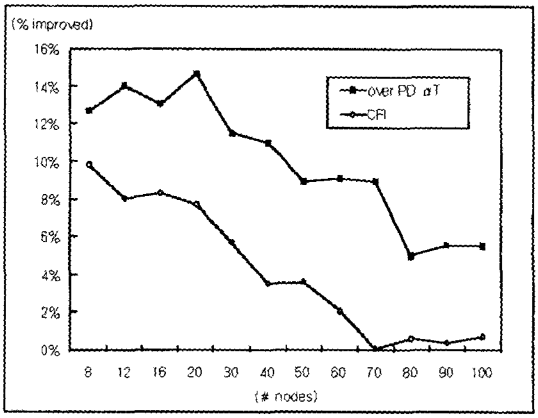


Figure 5. Average improvement of the proposed solution over the existing solutions

compared with that of Mosheiov's algorithm. More specifically, Mosheiov[9] reported that a random problem of 200 nodes required 263 seconds on IBM PC with 80486 processor, while the proposed algorithm took 205 seconds to solve the problems of 100 nodes (which is equivalent to 200 node problem for Mosheiov's algorithm since it regards one node as the two different nodes at the same location) on Pentium PC.

5. CONCLUDING REMARKS

In this paper, we proposed a heuristic algorithm for the vehicle routing problem with backhauls in which nodes may have both delivery and pick-up demands. The proposed algorithm basically finds a feasible tour by visiting the nodes with more delivery than pick-up (V^+), followed by visiting the nodes with more pick-up than delivery (V^-). Subsequently, it continues to improve the total travel distance by selecting the insertion with maximum improvement, while maintaining the feasibility. It is observed that finding the optimal tours for V^+ and V^- for an initial feasible tour does not always result in the better solution. That is, using nearest neighbor heuristic to solve V^+ and V^- for initial feasible tour results in good enough solutions.

The magnitude of error over the optimum solution can be dramatically reduced by using four initial solutions and select the best result (multiple run). Errors are reduced, by only about 1% as pattern exchanges are performed.

Larger problems with up to 100 nodes are tested by the proposed algorithm (without pattern exchanges). Computational results of solving random problems indicate that the proposed algorithm outperforms the two existing method of Mosheiov by 10% to 6% for practical range of the number of nodes within reasonable computation time.

As a future research, we can consider a branch and bound algorithm for the Traveling Salesman Problem which generates the n best solutions instead of single optimal solution. As the feasibility of n best solutions are examined in ascending order of the objective function value, the optimal solution for the VRPB studied in this paper is the tour which satisfies the feasibility for the first time.

REFERENCES

[1] Anily, S., "The Vehicle-Routing Problem with Delivery and Back-Haul Options," Naval Research Logistics 43, (1996), pp.415-434

[2] Bodin, L., Golden, B., Assad, A., and Ball, M., "Routing and scheduling of vehicle and crews - State of the art," Computers & Operations Research 10(2), 1983

[3] Casco, D., Golden, B., and Wasil, E., "Vehicle Routing with Backhauls: Models, Algorithms and case studies" , in Golden, B.L. and Assad, A.A. (eds.), *Vehicle Routing: Methods and Studies,* Studies in Management Science and Systems 16, North-Holland, Amsterdam, 1988

[4] Dief, I. and Bodin, L., "Extension of the Clarke and Wright Algorithm for Solving the Vehicle Routing Problem with Backhauling," Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management (A. E. Kidder, editor), Babson Park, MA, (1984), pp.75-96

[5] Fresh Air Fund "The Fresh Air Fund Annual Report 1989" , The Fresh Air Fund, 1040 Avenue of the Americas, New York, NY10018, 1989

[6] Goetschalckx, M. and Jacobs-Blecha, C., "The Vehicle Routing Problem with Backhauls," European Journal of Operations Research 42 (1989), pp.39-51

[7] Golden, B.L. and Assad, A.A. (eds.), *Vehicle Routing: Methods and Studies,* Studies in Management Science and Systems 16, North-Holland, Amsterdam, 1988

[8] Laporte, G., "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms," European Journal of Operations Research 59 (1992), pp.231-247

[9] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B., *The Traveling Salesman Problem: A Guided Tour of Combinatorial*

Optimization, Wiley, Chichester, 1985

[10] Mosheiov, G., "The Traveling Salesman Problem with Pick-up and Delivery" , European Journal of Operational Research 79 (1994), pp.299-310

[11] Yano, C., Chan, T., Richter, L., Cutler, T., Murty, K., and McGettigan, D., "Vehicle Routing at Quality Stores," Interfaces, 17(2), (1987), pp.52-63