

Nearest L -Neighbor Method with De-crossing in Vehicle Routing Problem

Hwan-Seong Kim[†] · Hoang-Son Tran-Ngoc*

[†] Department of Logistics, Korea Maritime University, Pusan 606 791, Republic of Korea

* Graduate School, Korea Maritime University, Pusan 606-791, Republic of Korea

Abstract : *The field of vehicle routing is currently growing rapidly because of many actual applications in truckload and less than truckload trucking, courier services, door to door services, and many other problems that generally hinder the optimization of transportation costs in a logistics network. The rapidly increasing number of customers in such a network has caused problems such as difficulty in cost optimization in terms of getting a global optimum solution in an acceptable time. Fast algorithms are needed to find sufficient solutions in a limited time that can be used for real time scheduling. In this paper, the nearest L-method (NLNM) is proposed to obtain a vehicle routing solution. String neighbors of different lengths were chosen, tested and compared. The applied de crossing procedure is meant to solve the routes by NLNM by giving a better solution and shorter computation time than that of NLNM with long string neighbors.*

Key words : *Nearest L-Neighbor Method, Vehicle Routing Problem, Logistics Network, De-crossing*

1. Introduction

Vehicle routing problem (VRP) is the generalization of the traveling salesman problem (TSP), which is to find the shortest possible tour to make exact single visits to each location... The VRP searches plausible paths or routes from a depot to customer locations for a fleet of capacitated vehicles to serve customers (pick up or/and delivery of goods or commodities) based on optimizing objective functions that indicate benefits (to maximize) or total cost and time (to minimize) of services.

The VRP has been the subject of many studies throughout the decade, as reviewed in (Gendreau et al, 1996; Ichoua et al., 2003; Haghani and Jung, 2005; Hashimoto et al., 2006; Fabri and Recht, 2006; Hanshar and Ombuki-Berman, 2007). The VRP has had important points in the scheduling of the routes of vehicles that carry materials, goods, products in a logistics network or chain of suppliers - manufacturers - warehouses and distribution centers - customers (Simchi-Levi et al., 2003), such as door-to-door services, courier services, full truckload (FTL) and less-than-truckload (LTL) services, etc. Recent applications of VRP in the fields of container terminals have been applied to the delivery of containers (Shin and Oh, 2008) or to the planning of a real-time location system (RTLS) (Shin et al., 2008), and so on.

We attempted to obtain a VRP solution to find routes that minimize the objective function at every location of vehicles in real time corresponding to the changing information

regarding new incoming customers and traffic.

This planning step is similar to control rules applied to a fleet of vehicles, such as actuators, in stages of the supply chain or logistics network, similar to a controlled system. The purpose is to find an optimal control rule that minimizes the operating cost of actuators or minimizes the transportation cost of vehicles in the VRP. The control rules should change over the operating time to react with the changes in environment (events, traffic condition, and etc.) and generate suitable decisions that optimize vehicle operation.

The planning routes for vehicles can be scheduled off-line for long term business or on-line for tactical business. We can see that more than 80 per cent of the operations of a logistics network is related to vehicle movements, such as vehicles traveling between stages of supply chain, and vehicles moving inside each stage of the supply chain. The control of these movements can enormously affect total cost. Consequently, a strategy in giving routes solution in VRP is very important.

However, it has been known that VRP is NP-hard. Because of its characteristics, VRP requires many techniques in finding an exact solution, especially by heuristic means. Differences in meta-heuristic methods are discussed by Michalewicz (1996); Toth and Vigo(2002); Chitty et al. (2004); Zaimpekis et al. (2005); Montemanni et al. (2005), and Fan et al. (2006).

Regarding previous works in this field, our main contribution is the generalization of the nearest neighbor

[†] Corresponding author: Hwan-Seong Kim, kimhs@hhu.ac.kr 051)410-4334

* tnhson1980@yahoo.com 051)410-4334

method by choosing a best group of L locations, which are not chosen to be served near the current position of the current vehicle to which the total cost is minimized.

Actually, when L reaches all existent requests N at the first step and if N is sufficiently large (>30), it might take much computational time to sort out from $N!$ possible permutations in order to obtain the best route solution (exact solution). Hence, the use of heuristic methods is needed to solve this case. In our method, a heuristic way is performed in two steps. First, choosing the nearest (least cost) number of locations (L) in the un-served customers set to get the local optimum route, and continuously grouping and optimizing the chosen group until all the locations are routed. Second, re-arranging the obtained route by moving along the route and repairing the "crosses" that appear, until no more crosses are detected. This procedure is called de-crossing.

In this paper, testing the nearest L -neighbor method (NLNM) with different values of length L , and also enhancing the route solutions with the de-crossing procedure to choose the best VRP solutions are presented. The results will show that NLNM with the support of the de-crossing procedure gives a better solution than applying only NLNM, and in many tests, it even reached an approximate or exact solution.

In section 2, the formulation of the vehicle routing problem is presented. The nearest L -neighbor method and de-crossing procedure are developed in section 3. Computational results in section 4 provide an indication of the benefits associated with the de-crossing procedure. Conclusion follows in section 5.

2. Vehicle Routing Problem

The basic formulation of objective function for VRP might be referred to Ahuja et al. (1993). Another formulation has been suitable for binary linear programming referred by Haghani et al. (2005). The former is very general but it does not consider the operational statuses of vehicles... The latter contains too many variables needed to be inputted in the decision solutions, and additionally it is very complex in formulating the problem and not suitable for VRP with a large number of customers. With the consideration of the operational vehicle's status and simple objective function, in this paper, the following simple objective function will be suggested intuitively:

2.1 Definitions

The vehicle routing problem is formally considered as a complete graph $G=(V,E)$, where $V=\{i \mid i=\overline{0,N}\}$ is the vertex set and $E=\{(i,j) \mid i \neq j, i=\overline{0,N}, j=\overline{0,N}\}$ is the edge set. Vertices $i=\overline{1,N}$ corresponding to customers with N as number of customers, whereas vertex 0 is the depot. A non-negative travel time t_{ij} is associated with the each edge $(i,j) \in E$.

Each customer i is characterized by a pickup location, a service time s_i , a time window $[e_i, l_i]$ and a vehicle planned arrival time t_i . If $t_i < e_i$, the vehicle has to wait up to e_i before servicing the customer and if $t_i > l_i$, the penalty is incurred in the objective.

The depot is characterized by a location, a time window $[e_0, l_0]$ for vehicle arrivals and departures, as well as the vehicle return time t_0^k for each vehicle $k \in K$, where K is the set of vehicles. The service time at the depot is assumed to be $s_0=0$. Each vehicle travels along a single route that starts and ends at the depot. The depiction of VRP is shown in Fig. 1.

The notation i_j^k is the customer in j th location serviced by k th vehicle. $[e_{i_j^k}, l_{i_j^k}]$ is the time window, and $s_{i_j^k}$ is the service time of customer i_j^k in the serving list of the vehicle k . And m_k is the number of customers that vehicle k will have been serving.

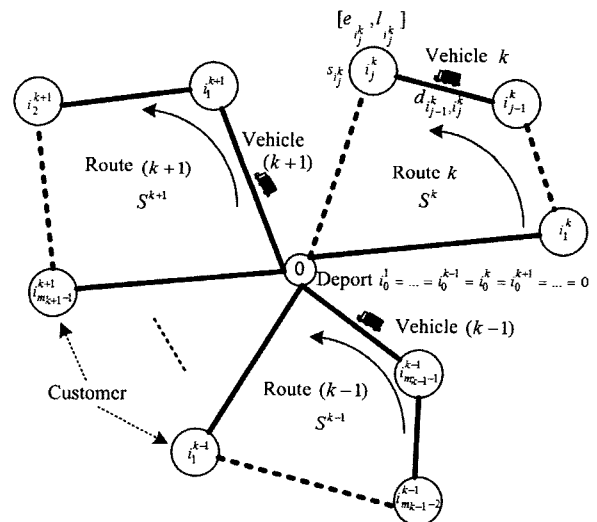


Fig. 1 Vehicle routing problem

2.2 Objective Function

The objective of VRP is to minimize the weighted summation of travel time, sum of waiting time at customer locations, sum of delay time at customer locations and delay time to return to the depot, which formulated in Eq. (1) for over all vehicles.

Assume the solution $S = \bigcup_{k \in K} S^k$, where $S^k = \{i_0^k, i_1^k, \dots, i_{m_k}^k\}$ is the sequence of customer locations visited by vehicle k with $i_0^k = i_{m_k}^k = 0$, then the objective function can be expressed as follows:

$$f(S) = \sum_{k \in K} f(S^k) = \sum_{k \in K} \left(a_1 \sum_{j=1}^{m_k} t_{i_{j-1}^k, i_j^k} + a_2 \sum_{j=1}^{m_k-1} (e_{i_j^k} - t_{i_j^k}^a)^+ + a_3 \sum_{j=1}^{m_k-1} (t_{i_j^k}^a + s_{i_j^k} - l_{i_j^k})^+ + a_4 (i_0^k - l_0)^+ \right) \quad (1)$$

where,

a_1, a_2, a_3, a_4 Weighting parameters.

$(\bullet)^+ = \max\{0, \bullet\}$ ex) $(x - y)^+ = \max\{0, x - y\}$

$t_{i_j^k}^a$ Arriving time to customer i_j^k of vehicle k .

$t_{i_j^k}$ Finished time at customer i_j^k of vehicle k and ready to move to next customer.

$t_{i_{j-1}^k, i_j^k} = t_{i_j^k}^a - t_{i_{j-1}^k}$

$\sum_{j=1}^{m_k} t_{i_{j-1}^k, i_j^k}$ Sum of waiting time at customer locations.

$\sum_{j=1}^{m_k-1} (e_{i_j^k} - t_{i_j^k}^a)^+$ Sum of delay time at customer locations.

$\sum_{j=1}^{m_k-1} (t_{i_j^k}^a + s_{i_j^k} - l_{i_j^k})^+$ Delay time to come back depot.

$(i_0^k - l_0)^+$ Sum of travel time.

If there is no waiting cost of vehicles to depot, then $(e_{i_{m_k}^k} - t_{i_{m_k}^k}^a)^+ = 0$ or $e_{i_{m_k}^k} \leq t_{i_{m_k}^k}^a$. Also the ready time in time

windows at depot for all vehicles will be zero ($e_{i_{m_k}^k} = 0$), and the travel time of each vehicle is always non-negative value ($t_{i_{m_k}^k} \geq 0$).

Moreover, if the waiting cost of vehicles on their return to depot and on servicing are identical ($a_4 = a_3$), then the objective function can be reduced as follows:

$$f(S) = \sum_{k \in K} f(S^k) = \sum_{k \in K} \sum_{j=1}^{m_k} g(S_j^k) = \sum_{k \in K} \sum_{j=1}^{m_k} \left(a_1 t_{i_{j-1}^k, i_j^k} + a_2 (e_{i_j^k} - t_{i_j^k}^a)^+ + a_3 (t_{i_j^k}^a + s_{i_j^k} - l_{i_j^k})^+ \right) \quad (2)$$

There are three serving situations at a customer of each vehicle that makes the change in the cost: waiting, normal and delay cases. A detailed objective function of each case will be expressed based on Eq. (2) as follows:

1) Waiting case

When the vehicle arrives at the customer location before the ready time (the time the customer needs to be served), then the vehicle has to wait for the right time (the customer accepts the receiving service), as shown in Fig. 2. The real total time serving this customer, $T_{i_{j-1}^k, i_j^k}$, and the cost for this link, $g(S_j^k)$, in this case will be given as follows:

$$T_{i_{j-1}^k, i_j^k} = t_{i_j^k} - t_{i_{j-1}^k} = e_{i_j^k} - t_{i_{j-1}^k} + s_{i_j^k} \quad (3)$$

$$g(S_j^k) = a_1 t_{i_{j-1}^k, i_j^k} + a_2 (e_{i_j^k} - t_{i_j^k}^a)^+ \quad (4)$$

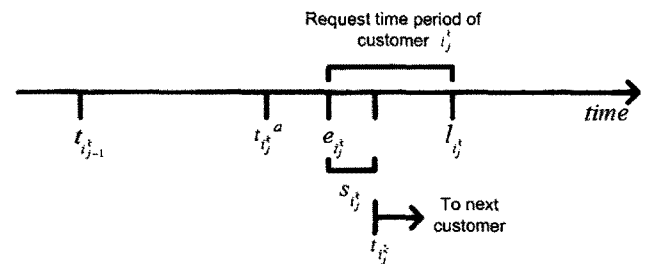


Fig. 2 Serving situations: waiting case

2) Normal case

Normal case or the right time case is shown in Fig. 3. There is no penalty in the cost link. The real total spent servicing time of the customer t_j^k and the cost function for this link will be calculated as follows:

$$\begin{aligned} T_{t_{j-1}^k, t_j^k} &= t_j^k - t_{j-1}^k \\ &= t_j^{k,a} - t_{j-1}^k + s_{ij}^k \end{aligned} \quad (5)$$

$$g(S_j^k) = a_1 t_{j-1, j}^k \quad (6)$$

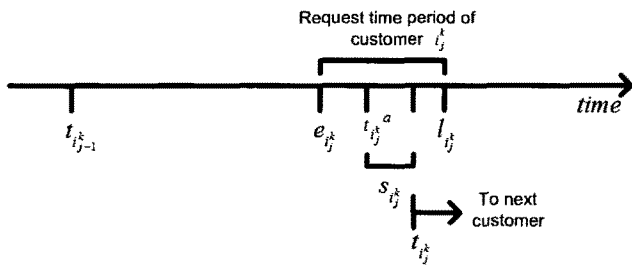


Fig. 3 Serving situations: normal case

3) Delay cases

Slowing cases are showed in Figs. 4 and 5, respectively. In Fig. 4, the vehicle arrives in a valid period time, but the servicing time exceeds the due time. In Fig. 5, the vehicle reaches the customer totally late. There is a penalty for this lateness. The real total time servicing this customer, $T_{t_{j-1}^k, t_j^k}$, and the cost for this link, $g(S_j^k)$, in this case will be calculated as follows:

$$\begin{aligned} T_{t_{j-1}^k, t_j^k} &= t_j^k - t_{j-1}^k \\ &= t_j^{k,a} - t_{j-1}^k + s_{ij}^k \end{aligned} \quad (7)$$

$$g(S_j^k) = a_1 t_{j-1, j}^k + a_3 (t_{ij}^{k,a} + s_{ij}^k - l_{ij}^k) \quad (8)$$

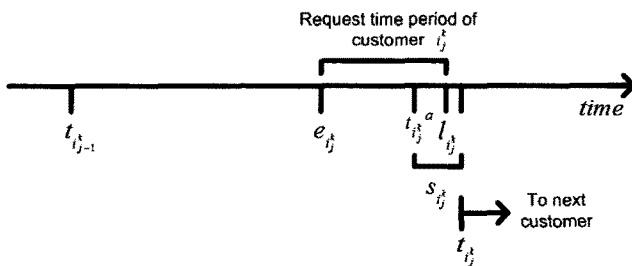


Fig. 4 Serving situations: delaying case 1

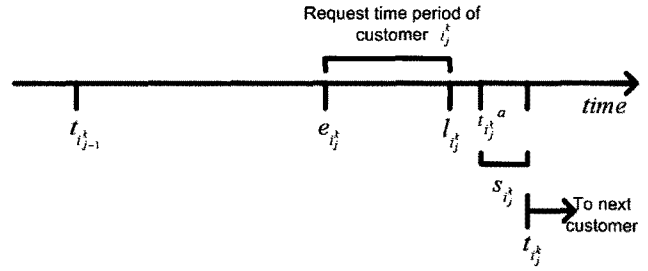


Fig. 5 Serving situations: delaying case 2

3. Nearest L - Neighbor Method with De-crossing

3.1 Nearest L -Neighbor Method

The nearest neighbor method (NNM) is known as a technique for finding the closest point in metric spaces (Arya et al., 1994). A generalization of the nearest neighbor method (NNM) is used to plan vehicular routes. In the nearest neighbor method, at the current step, only one nearest customer location in all remaining un-served customers is chosen, whereas in nearest L - neighbor method (NLNM), a sequence of L customers is selected to ensure the least total cost.

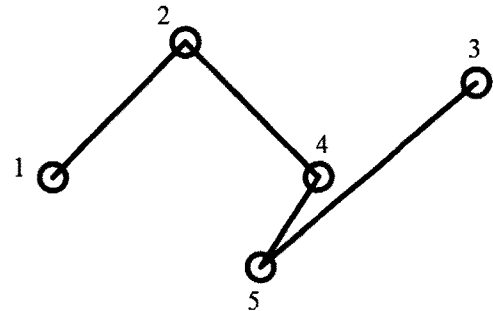


Fig. 6 Solution of NNM

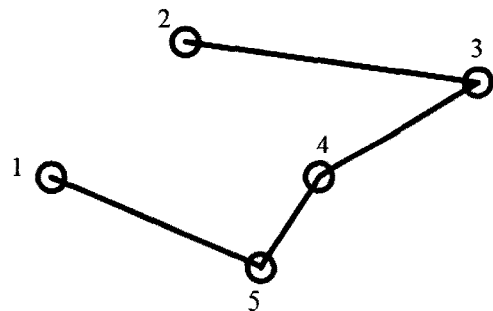


Fig. 7 Solution of NLNM with $L=2$

As an example, Figs. 6 and 7 show the solutions of NNM and NLNM with $L=2$, respectively, for five customers located in Euclidean space. The solution in Fig. 7 looks smoother and shorter than that in Fig. 6.

Therefore, NNM is a special case of NLNM when $L=1$ along selecting progress. When L equals to the total number of customers, then NLNM gives an exact solution. However, when the number of customers is sufficiently large as in real applications, the time to get the exact solution is unacceptable. At a time we plan a solution for a group of customers, and continuously we plan for other groups until all the locations are considered.

For a graph of N vertexes, NNM needs $C = N(N+1)/2$ comparison steps to get a suboptimum solution.

In case of NLNM, it needs:

$$C = \frac{N!}{(N-L)!} + \frac{(N-L)!}{(N-2L)!} + \dots + \frac{(N-(n-1)L)!}{(N-nL)!} + (N-nL)! \quad (9)$$

$$n = \left\lfloor \frac{N}{L} \right\rfloor \quad (10)$$

where $M! = M \times (M-1) \times \dots \times 2 \times 1$ and $\lfloor \cdot \rfloor$ means the greatest integer in the argument.

In Eq. (9), let $L \rightarrow N$, then $C \rightarrow N!$. This is the total number of elements in solution space. Theoretically, we could check all the elements of this space to get the exact solution.

Let us give a set V customers and define that $A^L(V)$ is all sets of L customers in set V , and ${}^uS^k$ is set of customers served or will be served of vehicle k at time u by NLNM. Then, we have:

$$A_{\min}^L(V) = \min_f [A^L(V)] \quad (11)$$

$${}^uS^k = ({}^{u-1}S^k \cup A_{\min}^L(V-S)) \quad (12)$$

Eq. (11) and Eq. (12) are useful in updating the routes in static VRP.

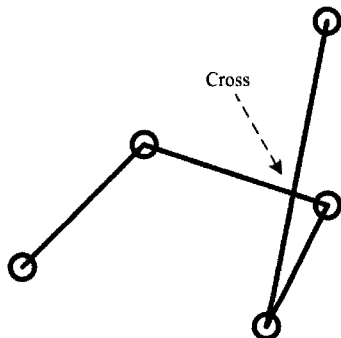


Fig. 8 Cross in the route from NLNM

NLNM sometimes gives a solution with crosses, as shown for example in Fig. 8 which makes the route look like a bad solution. The de-crossing procedure in the next subsection is to remove the crosses and reduce the length of the route. Therefore, it enhances the quality of the solution.

3.2 De-Crossing Procedure

Assume a crossing situation as a part of a solution called $S_{cross} = \{V_1, V_2, V_3, V_4\}$ which is shown in Fig. 9.

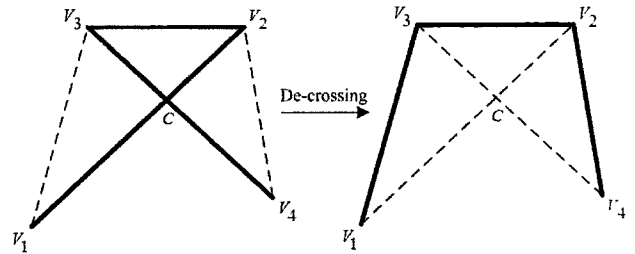


Fig. 9 Crossing $\{V_1, V_2, V_3, V_4\}$ and de-crossing $\{V_1, V_3, V_2, V_4\}$

The intersect point is $(V_1, V_2) \cap (V_3, V_4) = \{C\}$.

In Fig. 9, the following inequality equations are induced.

$$d_{V_1, C} + d_{C, V_3} \geq d_{V_1, V_3} \quad (13)$$

$$d_{V_2, C} + d_{C, V_4} \geq d_{V_2, V_4} \quad (14)$$

By adding Eq. (13) and Eq. (14), we have

$$d_{V_1, C} + d_{C, V_3} + d_{V_2, C} + d_{C, V_4} \geq d_{V_1, V_3} + d_{V_2, V_4} \quad (15)$$

Also by reducing Eq. (15), Eq. (16) will be obtained

$$d_{V_1, V_2, V_3, V_4} \geq d_{V_1, V_3, V_2, V_4} \quad (16)$$

Inequality Eq. (16) implies that the changing order of $\{V_1, V_2, V_3, V_4\}$ to $\{V_1, V_3, V_2, V_4\}$ by swapping (or de-crossing as in Fig. 10) $\{V_2\}$ and $\{V_3\}$ that reduces the distance from $\{V_1\}$ to $\{V_4\}$, and consequently, it reduces cost. That is the effect of the de-crossing procedure.

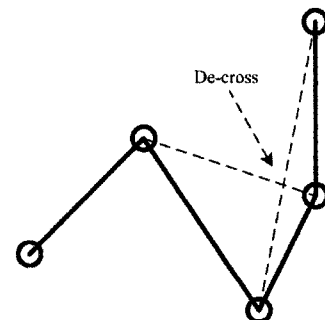


Fig. 10 De-crossing of example in Fig. 8

By applying the de-crossing procedure in Fig. 8, we have a de-crossing result route as shown in Fig. 10.

In general, the d_{V_1, V_x, V_y, V_4} in inequality Eq. (16) is not only the distance from $\{V_1\}$ to $\{V_4\}$ in Euclidian space but also is the value from the objective function to the set $\{V_1, V_x, V_y, V_4\}$.

4. Simulation Results

4.1 Simulation Conditions

To verify the proposed NLNM, Solomon data from reference [15] were used. Among Solomon data samples, 100 customers with a different spatial distribution of location are selected, where its schedule for serving includes ready time, due date, service time, and capacities are considered. Also, Solomon data have three kinds of samples R, C, and RC.

R(random) sampledistributes the customer location randomly C (clustered) sample clusters the customer location in well-defined geographic cluster RC samples consider the customer location of both cases (random and clustered).

For simulation, we used a Pentium IV(2GHz clock speed) computer with 1GB DDRAM. To prove the efficiency of the NLNM and NLNM with de-crossing, three cases of simulation conditions are considered as shown in Table 1.

Table 1 Simulation cases

Cases	a_1	a_2	a_3	Comments
Case 1	1	0	0	Only consider travel times
Case 2	1	0	1	Consider travel times and cost of delay serving
Case 3	1	1	1	Consider travel times and cost of waiting and delay serving

4.2 Simulation Results

1) NLNM and Effects of De-crossing in NLNM

To verify the NLNM and the effectiveness of de-crossing in NLNM in this paper, nine customers with single vehicle was considered. Tables 2and 3 show the results of NLNM and NLNM with de-crossing, respectively. The decreased rate of cost is shown by increasingthe number of neighbor L in Figs. 11 and 12, respectively.

As shown in Table 2 and Fig. 11, the cost will be decreased by increasing the number of neighbor L . However, the exact solution is not guaranteed to be obtained by NLNM. However, NLNM with de-crossing can quickly find a near-exact solution as shown in Table 3 and Fig. 12.

Table 2 Simulation results of VRP

N=9	Simulation cases					
	Case 1		Case 2		Case 3	
L	cost	time(s)	cost	time(s)	cost	time(s)
1	158	0.01	1086	0.01	1182	0.02
2	176	0.04	967	0.04	1351	0.03
3	158	0.14	1197	0.14	992	0.14
4	146	0.68	1104	0.68	1079	0.68
5	146	3.71	954	3.58	984	3.57
6	158	15.12	940	15.02	969	15.05
7	158	48.18	835	48.57	895	48.55
8	161	103.76	714	103.26	732	103.40
9	146	110.24	667	110.11	685	110.26

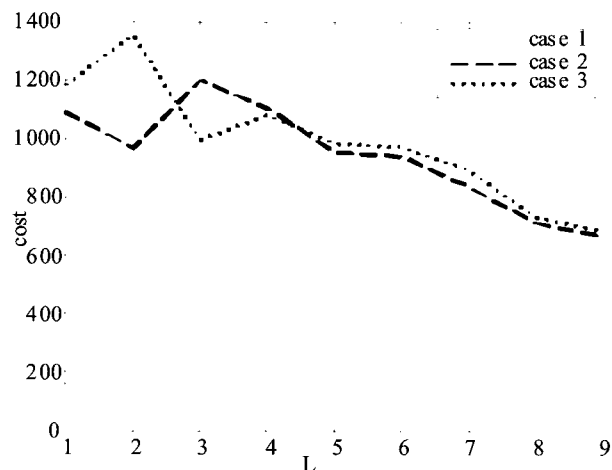


Fig. 11 Simulation results of NLNM

Table 3 Simulation results of VRP with de-crossing

N=9	Simulation cases					
	Case 1		Case 2		Case 3	
L	cost	time(s)	cost	time(s)	cost	time(s)
1	146	0.03	771	0.02	795	0.02
2	151	0.05	819	0.05	852	0.05
3	146	0.14	902	0.15	795	0.15
4	146	0.70	737	0.71	851	0.70
5	146	3.58	667	3.64	685	3.58
6	146	15.22	667	15.24	685	15.25
7	146	49.08	737	49.10	732	49.00
8	149	104.65	714	104.30	732	104.34
9	146	111.30	667	111.58	685	110.77

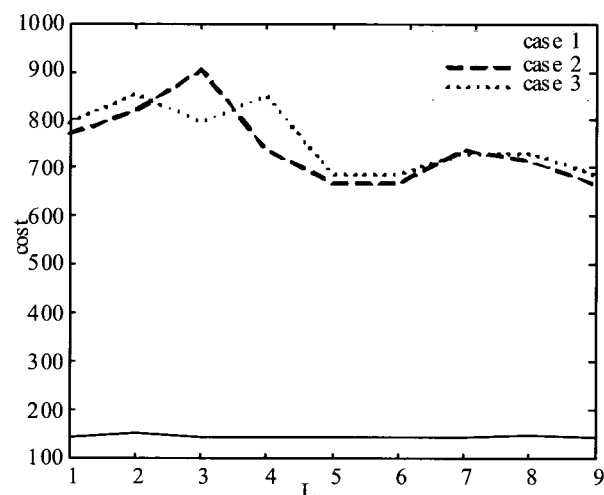


Fig. 12 Simulation results of NLNM with de-crossing

2) Simulation Results for Single Vehicle

To verify the proposed NLNM in many customer cases, we will consider 100 customers random sample (R101) with a single vehicle. The simulation results are given in Figs. 13 - 18 for three cases without/with de-crossing. The cost and computational time of simulation results are summarized in Table 4.

Figs. 13, 15 and 17 show the results of NLNM without de-crossing for 100 customers. Figs. 14, 16 and 18 show the results of NLNM with de-crossing.

Fig. 14 shows a completely free crossing in the case in which the vehicle visits all customers and returns to depot. In Figs. 16 and 18, several crosses appear in the routes because the re-arrangement depended on the objective function in consideration of penalties. So, the de-crossing can reduce the cost space, not in ordinary Euclidean space.

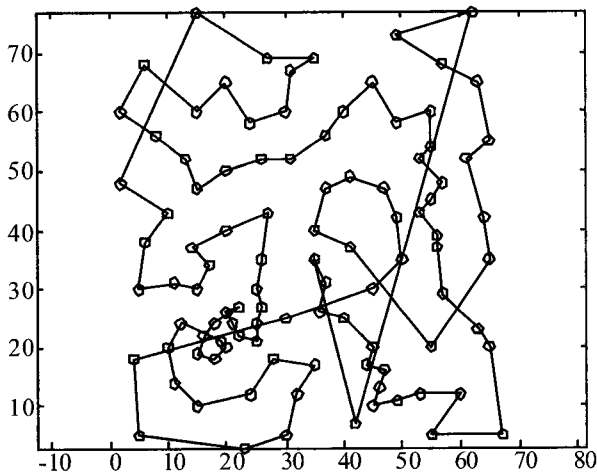


Fig. 13 VRP without de-crossing of sample R101 with $N=100$, $K=1$ and $L=1$, case 1

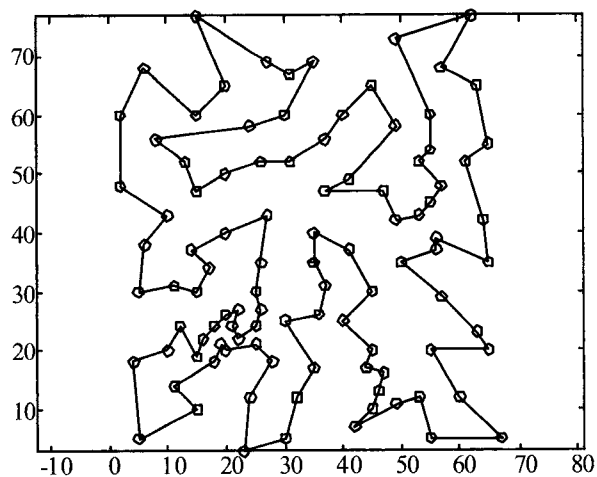


Fig. 14 VRP with de-crossing of sample R101 with $N=100$, $K=1$ and $L=1$, case 1

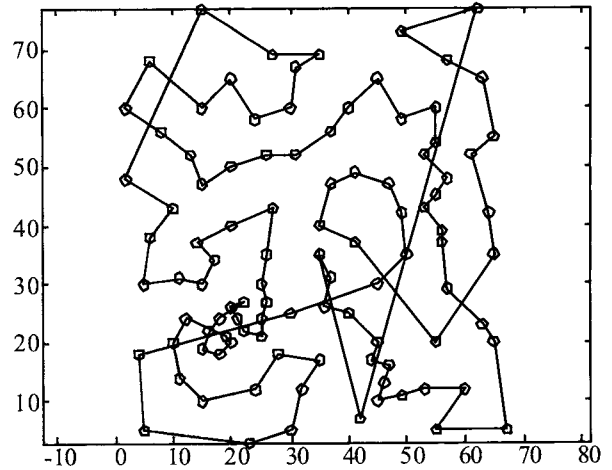


Fig. 15 VRP without de-crossing of sample R101 with $N=100$, $K=1$ and $L=1$, case 2

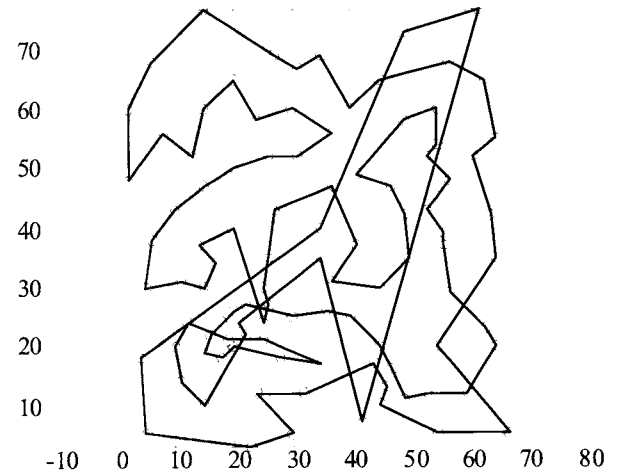


Fig. 16 VRP with de-crossing of sample R101 with $N=100$, $K=1$ and $L=1$, case 2

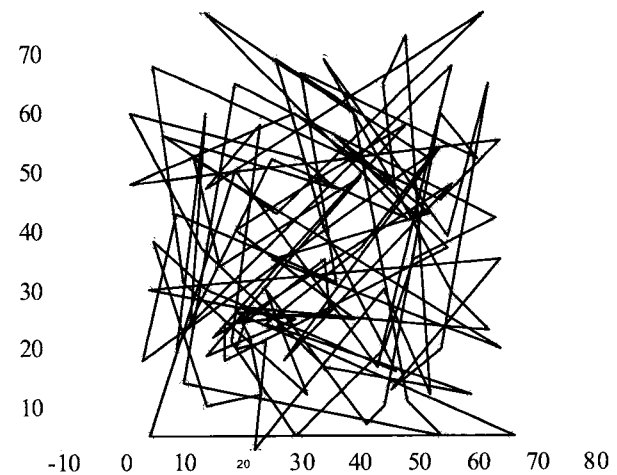


Fig. 17 VRP without de-crossing of sample R101 with $N=100$, $K=1$ and $L=1$, case 3

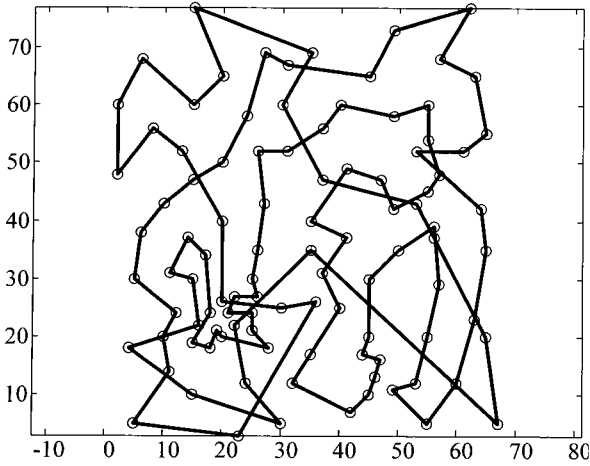


Fig. 18 VRP with de-crossing of sample R101 with $N=100$, $K=1$ and $L=1$, case 3

Table 4 Simulation results of VRP with single vehicle

N=100	Simulation cases					
	Case 1		Case 2		Case 3	
	cost	time(s)	cost	time(s)	cost	time(s)
without	808	1.14	93,205	1.02	200,100	1.01
with de-crossing	678	33.03	76,699	28.50	81,091	37.91

In Table 4, the cost of NLNM with de-crossing is reduced more than that of NLNM. So, the NLNM with de-crossing is a more powerful means to get a better solution than NLNM. However, the comparison results with exact solution could not be shown in this paper because our computer took a long time in the calculation of simulation.

3) Simulation Results for Multi Vehicle

The simulation of multiple vehicles is applied to 100 customers. To show the effectiveness of NLNM with de-crossing in the multi-vehicle situation (case 1), $L=1$ is

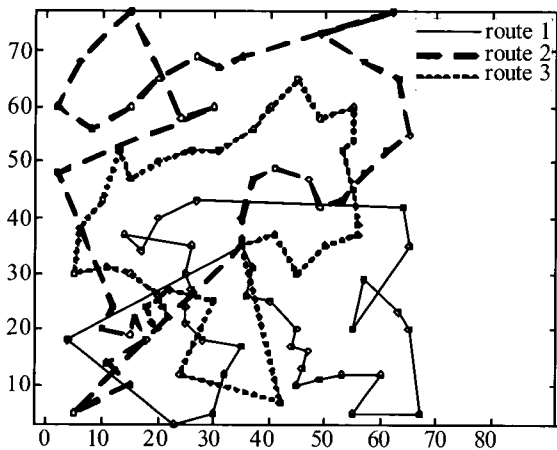


Fig. 19 VRP with sample R101 with $N=100$, $K=3$ and $L=1$, Case 1

considered and the results are shown in Figs. 19 and 20, where they are used. Moreover, the cost and computational time of simulation results with multiple vehicles are summarized in Table 5 for three cases with changed vehicle.

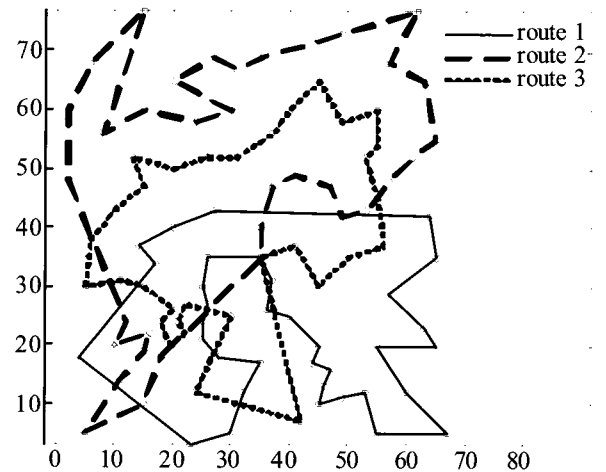


Fig. 20 VRP with de-crossing of sample R101 with $N=100$, $K=3$ and $L=1$, Case 1

Table 5 Simulation results of VRP with multi vehicle

k	N=100	Simulation cases					
		Case 1		Case 2		Case 3	
		cost	time(s)	cost	time(s)	cost	time(s)
1	without	808	1.14	93,205	1.02	200,100	1.01
	with de-crossing	678	33.03	76,699	28.50	81,091	37.91
2	without	624	1.09	134,360	0.98	204,250	0.98
	with de-crossing	555	57.14	116,980	98.88	120,350	98.69
3	without	556	1.19	177,090	1.00	241,010	0.99
	with de-crossing	509	164.91	155,430	143.80	161,530	250.48

As a result, the cost of the objective function can be reduced by using the NLNM with de-crossing more than NLNM, but the computation of NLNM with de-crossing takes a longer time than that of NLNM.

5. Conclusion

In this paper, the NLNM method has been proposed to find a suboptimal routing solution for the vehicle routing problem involving multiple vehicles. String neighbors of different lengths were chosen, tested and compared. Also, the de-crossing procedure applied to solution routes from NLNM gives a better solution and takes a shorter computation time than solutions involving long string neighbors from NLNM.

The proposed method will be useful for adaptation to real vehicle routing problems with real-time scheduling. Also, the proposed method will be applied to the Dynamic Vehicle Routing Problem (DVRP) in the future.

References

- [1] Ahuja, R.K., Magnanti, T.L., and Orlin J.B.(1993), "Network flows: theory, algorithms and applications", 1st edition, Prentice Hall.
- [2] Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. and Wu, A. Y. (1994), "An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions", *Journal of the ACM*, Vol. 45, No. 6, pp. 891-923.
- [3] Chitty, D.M. and Hernandez, M.L.(2004), "A hybrid ant colony optimization technique for dynamic vehicle routing", Book chapter, *Genetic and Evolutionary Computation - GECCO 2004*, pp. 48-59.
- [4] Fabri, A. and Recht, P.(2006), "On Dynamic Pickup and Delivery Vehicle Routing with Several Time Windows and Waiting Times", *Journal of Transportation Research Part B*, Vol. 40, pp. 335-350.
- [5] Fan, J., Wang, X., and Ning, H.(2006), "A Multiple Vehicles Routing Problem Algorithm with Stochastic Demand", *The Sixth World on Intelligent Control and Automation WCICA*.
- [6] Gendreau, M., Laporte, G., and Seguin R.(1996), "Stochastic vehicle routing", *European Journal Of Operational Research*, Vol. 88, pp. 3-12.
- [7] Haghani, A. and Jung, S.(2005), "A Dynamic Vehicle Routing Problem with Time-Dependent Travel Times", *Journal of Computers & Operations Research*, Vol. 32, pp. 2959-2986.
- [8] Hashimoto, H., Ibaraki, T., Imahori, S., and Yagiura, M.(2006), "The Vehicle Routing Problem with Flexible Time Windows and Travelling Times", *Journal of Discrete Applied Mathematics*, Vol. 154, pp. 2271-2290.
- [9] Hanshar, F.T. and Ombuki-Berman, B.M.(2007), "Dynamic vehicle routing using genetic algorithms", *Applied Intelligent*, Vol. 27, pp. 89-99.
- [10] Ichoua, S., Gendreau, M., and Potvin, J.Y.(2003), "Vehicle Dispatching with Time-Dependent Travel Times", *European Journal of Operational Research*, Vol. 114, pp. 379-396.
- [11] Michalewicz (1996), "Genetic algorithms + data structures = evolution programs", 3rd edition, Springer.
- [12] Montemanni, R. Gambardella, L.M., Rizzoli, A.E., and Donati, A.V.(2005), "Ant colony system for a dynamic vehicle routing problem", *Journal of Combinatorial Optimization*, Vol. 10, pp. 327-343.
- [13] Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E.(2003), "Design and Managing the Supply Chain: Concepts, Strategies and Case Studies", McGraw Hill.
- [14] Solomon data, <http://www.idsia.ch/~luca/macsvrptw/problems/welcometw.htm>
- [15] Shin, J.H., Chang, M.H., and Yu, S.J. (2008), "A Study on RTLS Technology based YT Dynamic Operation for Efficiency of Container Terminal", *Journal of Navigation and Port Research*, Vol. 32, No. 5, pp. 369-377.
- [16] Shin, J.H., Yu, S.J., and Chang, M.H. (2008), "A Study on Optimized Decision Model for Transfer Crane Operation in Container Terminal", *Journal of Navigation and Port Research*, Vol. 32, No. 6, pp. 465-471.
- [17] Shin, J.Y. and Oh, S.I. (2008), "Vehicle Routing Problem for delivering containers using combined chassis trailers", *Conference of 2008 KINPR*, pp. 155-156.
- [18] Toth, P. and Vigo, D.(2002), "The Vehicle Routing Problem", *Society for Industrial and Applied Mathematics (SIAM)*.
- [19] Zempeki, V. and Giaglis, G.M.(2005), "A Dynamic Real-Time Vehicle Routing System for Distribution Operations", *Consumer Driven Electronic Transformation*, Springer.

Received 23 February 2009

Revised 30 March 2009

Accepted 30 March 2009