Int J Adv Manuf Technol (2005) 26: 372–381
DOI 10.1007/s00170-003-2005-3

ORIGINAL ARTICLE

S.C. Liu · C.C. Lin

A heuristic method for the combined location routing and inventory problem

Received: 4 August 2003 / Accepted: 28 October 2003 / Published online: 1 December 2004

Abstract The combined location routing and inventory problem (CLRIP) is used to allocate depots from several potential locations, to schedule vehicles' routes to meet customers' demands, and to determine the inventory policy based on the information of customers' demands, in order to minimize the total system cost. Since finding the optimal solution(s) for this problem is a nonpolynomial (NP) problem, several heuristics for searching local optima have been proposed. However, the solutions for these heuristics are trapped in local optima. Global search heuristic methods, such as tabu search, simulated annealing method, etc., have been known for overcoming the combinatorial problems such as CLRIP, etc. In this paper, the CLRIP is decomposed into two subproblems: depot location-allocation problem, and routing and inventory problem. A heuristic method is proposed to find solutions for CLRIP. First of all, an initial solution for CLRIP is determined. Then a hybrid heuristic combining tabu search with simulated annealing sharing the same tabu list is used to improve the initial solution for each subproblem separately and alternatively. The proposed heuristic method is tested and evaluated via simulation. The results show the proposed heuristic method is better than the existing methods and global search heuristic methods in terms of average system cost.

Keywords Combined location routing and inventory problem (CLRIP) · Global search heuristic method · Heuristic method · Hybrid heuristic · NP problem

1 Introduction

The combined location routing and inventory problem (CLRIP) is to allocate depots from several potential locations, to schedule vehicles' routes to meet customers' demands, and to determine the inventory policy (such as order quantity during each production run, order-up-to level for replenishment, etc.) based on the information of customers' demands, in order to minimize the total system cost (including location, transportation, and inventory costs). CLRIP consists of three subproblems: depot location-allocation problem, vehicle routing problem, and inventory control problem. The decisions for solving these three subproblems are interrelated to one another. For example, when the order quantity during each production run decreases, transportation costs will increase and inventory costs will decrease [1]. Furthermore, the routing and inventory control decision will affect the decision of depot locations since locations are determined based on the minimal system cost criterion. Inversely, when the depot locations are determined, the vehicle routing and inventory policy will be affected and decided based on the minimal total system cost criterion. Hence, how to determine depot locations, vehicle routing, and inventory policy, becomes an important issue in designing distribution systems such as food and drink industries, delivery to retail shops, and distribution of various consumer goods [2, 3].

In this paper, we are trying to determine depot locations, vehicle routing, and inventory policy based on the minimal system cost criterion. As for the assumptions for vehicle routes, we adopt the following constraints: (1) the total demand for customers on any route is not greater than the vehicle service capacity, (2) the order quantity during each production run on any route is not greater than the vehicle capacity, and (3) each route is served by one vehicle. The model formulation for CLRIP is similar to that in Liu and Lee's research [3] (please refer to Appendix A for the details).

The location routing problem (LRP) composed of two subproblems: depot location-allocation problem and vehicle routing problem, has been shown to be NP-hard [4]. Since CLRIP is more complex than LRP, CLRIP belongs to the class of NP problems. Liu and Lee [3] proposed a heuristic method to find solutions for CLRIP. The initial solution using route-first, location-allocation second approach based on the minimal system cost is determined. Then an improvement heuristic search for a better solution based on the initial solution is developed.

S.C. Liu (✉) · C.C. Lin
Department of Management Information Systems
National Pingtung University of Science and Technology
Pingtung, Taiwan 912, R.O.C.
E-mail: sliu@mail.npust.edu.tw
Tel.: +886-8-7703202-7911
Fax: +886-8-7740306

However, the improvement search is a local-optimality search. To avoid being trapped in local optima, global search heuristic methods (such as tabu search, simulated annealing, etc.) have been used for solving the combinatorial problems [5]. However, there was no global search heuristic method proposed for solving CLRIP until now. Since the problem structure of LRP is similar to that of CLRIP, global search heuristic methods used in LRP are reviewed in this paper. Renaud et al. [6] proposed a tabu search algorithm for the multi-depot vehicle routing problem with capacity and route length restrictions. Three search strategies: fast improvement, intensification, and diversification are used in tabu search to improve effectiveness and efficiency. Tuzun and Burke [7] proposed a two-phase tabu search for the location routing problem. The two-phase approach coordinates two tabu search mechanisms – one seeking a good facility configuration, the other a good routing that corresponds to this configuration. Wu et al. [4] proposed a simulated annealing algorithm for solving the location routing problem. The problem is divided into two subproblems: the location-allocation problem and the vehicle routing problem. Each is solved by the simulated annealing algorithm with tabu lists to avoid cycling.

According to the review above, we conclude that CLRIP can be decomposed into two subproblems: depot location-allocation problem, and routing and inventory problem. Each subproblem can be solved by a hybrid heuristic combining tabu search (TS) with simulated annealing (SA) sharing the same tabu list, where TS and SA are two well-known methods to solve the combinatorial problems such as CLRIP. The reasons for choosing the hybrid heuristic approach are: (1) to share the same tabu list between TS and SA to avoid search cycling and improve search efficiency and (2) to improve search effectiveness.

2 The proposed heuristic method

In this paper, a heuristic method is proposed to solve CLRIP consisting of two subproblems: depot location-allocation problem, and routing and inventory problem. First, the heuristic method proposes an initial solution procedure for CLRIP. Then a hybrid heuristic combining TS with SA sharing the same tabu list is proposed to find solutions for each subproblem of CLRIP separately and alternatively. The procedure of this heuristic method is as follows (Fig. 1):

Obtaining the initial solution (Fig. 2)

Step 1. (1) Set $r = 1$, $k = 1$, MaxSup = vehicle service capacity, count = 1. (2) Put all customers into a set F. (3) Put all depots into a set E.

Step 2. (1) Randomly select a customer from F. (2) Put the customer into V_k. (3) Delete the customer from F. (4) Set $k = k + 1$.

Step 3. Is F empty? If yes, go to step 4. Otherwise, go to step 2.

Step 4. (1) Select a depot from E with the shortest path to the centroid of V_r. (2) Put the depot into V_r. (3) Set $r = r + 1$.

Step 5. Is $r > k$? If yes, go to step 6. Otherwise, go to step 4.

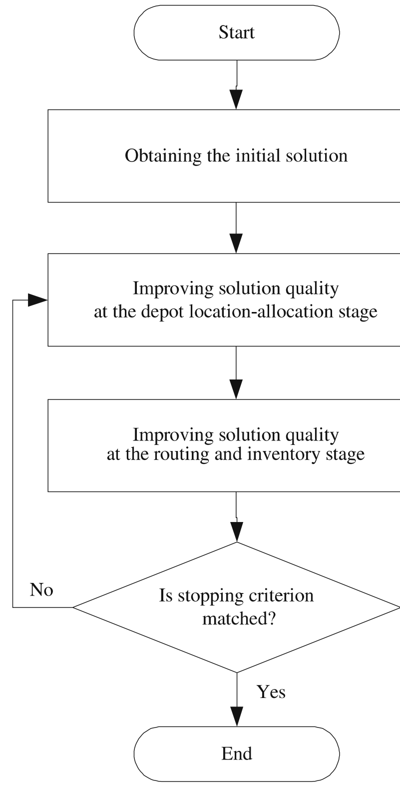


Fig. 1. The flowchart for the proposed heuristic method

Step 6. (1) Compute the total system cost SC (It can be computed based on the objective function mentioned in Appendix A.). (2) Set the initial solution SC and V_t as the temporary best solution (Let the best solution $X^* =$ the current solution X^0 (V_t for $1 \leqslant t \leqslant k$) and $SC(X^*) = SC(X^0)$).

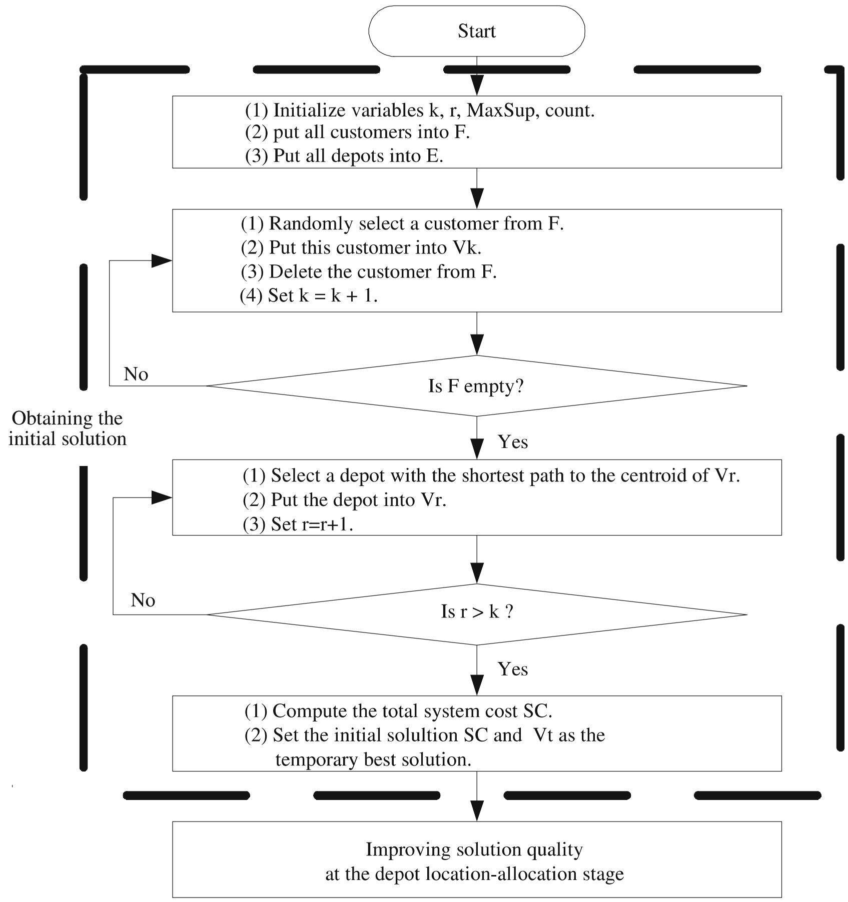
Improving solution quality at the depot location-allocation stage (Fig. 3)

Step 7. Generate a candidate move (from X^0 to the candidate solution X^1) using drop procedure or swap procedure randomly at the depot location-allocation stage.
The drop procedure is to randomly select an open depot D_i in a route V_i and find another open depot D_j that is nearest to the route V_i. Substitute D_i with D_j to serve the customers in V_i. Close D_i.
The swap procedure is to randomly select two open depots: D_i in V_i and D_j in V_j. Exchange D_i and D_j.

Step 8. Are the total demands of customers in V_i and V_j less than or equal to MaxSup, and the order quantities during each production run in V_i and V_j less than or equal to vehicle capacity? If yes, go to step 9. Otherwise, go to step 7.

Step 9. Is the candidate move in the tabu list (The length of tabu list is set equal to 7 [8].)? If yes, go to step 10. Otherwise, (1) update $X^0 = X^1$, $SC(X^0) = SC(X^1)$ (please refer to Appendix A for details), (2) update the tabu

Fig. 2. The flowchart for obtaining the initial solution



list at the depot location-allocation stage, and (3) go to step 11.

Step 10. Is $SC(X^1) \leqslant SC(X^*)$? If yes, (1) update $X^* = X^1$, $SC(X^*) = SC(X^1)$, (2) update $X^0 = X^1$, $SC(X^0) = SC(X^1)$, (3) update the tabu list at the depot location-allocation stage, and (4) go to step 11. Otherwise, go to step 7.

Step 11. Initialize the parameters for simulated annealing search such as initial temperature (= 70), reheating factor r (= 0.9), stopping temperature (= 10) [9].

Step 12. Generate a neighboring solution X^1 as the next candidate solution using drop procedure or swap procedure randomly (please refer to step 7 for details of these two procedures).

Step 13. Are the total demands of customers in V_i and V_j less than or equal to MaxSup and the order quantities during each production run in V_i and V_j less than or equal to vehicle capacity? If yes, go to step 12. Otherwise, go to step 14.

Step 14. Is the move (from X^0 to the neighboring solution X^1) in the tabu list? If yes, go to step 12. Otherwise, go to step 15.

Step 15. Is the neighboring solution accepted?
1. Let $\Delta SC = SC(X^1) - SC(X^0)$.
2. If $\Delta SC \leqslant 0$, then $X^0 = X^1$, $SC(X^0) = SC(X^1)$ and update the tabu list at the depot location-allocation stage. And if $SC(X^1) \leqslant SC(X^*)$, then $X^* = X^1$, $SC(X^*) = SC(X^1)$.
3. If $\Delta SC > 0$, then $X^0 = X^1$, $SC(X^0) = SC(X^1)$ with probability $\exp(-\Delta SC/T)$ and update the tabu list at the depot location-allocation stage.

Step 16. Should the procedure stop under the temperature T? If yes, go to step 17; otherwise, go to step 7.
When the number of accepted solutions under the temperature T reaches to a predefined value, the following condition should be checked:

$$\frac{\left|AVG_e - AVG_f\right|}{AVG_f} \leqslant \varepsilon \,,$$

where AVG_e is the average objective value of accepted solutions under the temperature T. AVG_f is the average objective value of accepted solutions before T. ε is a predefined equilibrium value ($0 < \varepsilon < 1$). If the above condition is satisfied, the equilibrium state has been reached and the procedure stops under T [10].

Step 17. $T = T \times r$.

Step 18. Is the stopping criterion ($T <$ stopping temperature) at the depot location-allocation stage matched. If yes, go to step 19; otherwise, go to step 7.

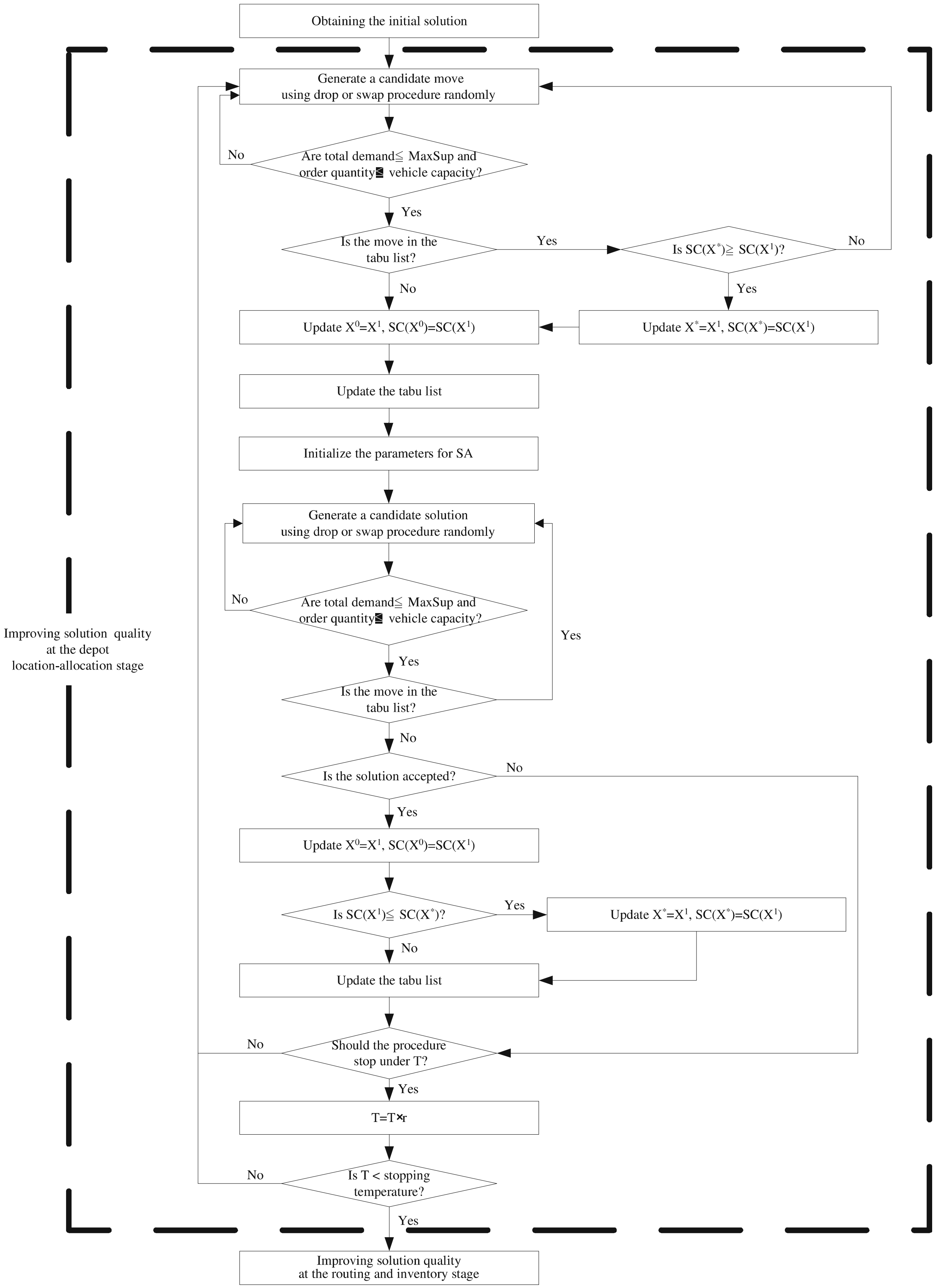


Fig. 3. The flowchart for improving solution quality at the depot location-allocation stage

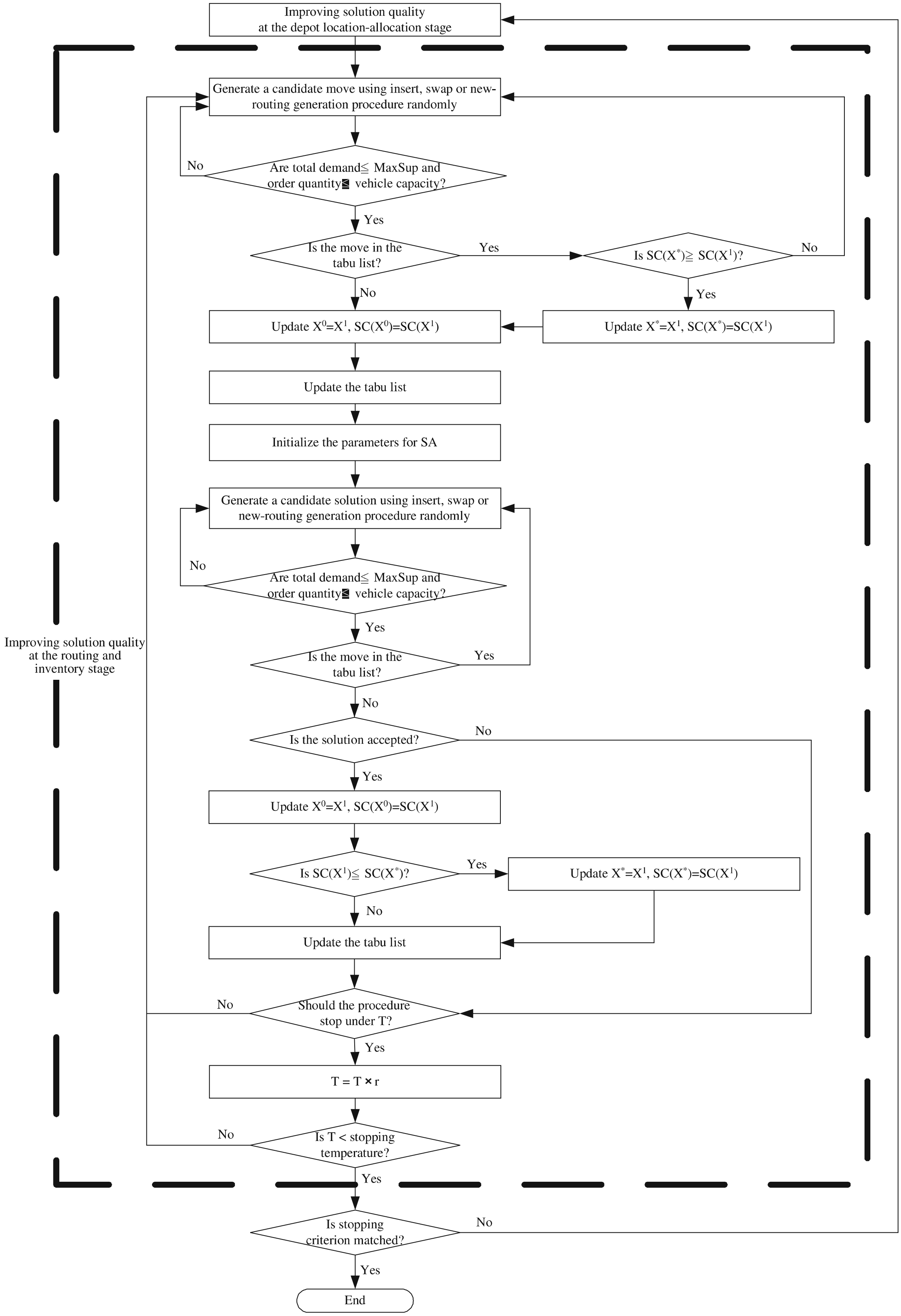


Fig. 4. The flowchart for improving solution quality at the routing and inventory stage

Improving solution quality at the routing and inventory stage (Fig. 4)

Step 19. Generate a candidate move (from X^0 to X^1) using the insertion procedure, swap procedure, or new-route generation procedure randomly at the routing and inventory stage.
The insertion procedure is to randomly select two routes: V_i and V_j. Then randomly select a customer C_1 in V_i. Find two customers, C_2 and C_3 in V_j that are nearest to C_1. Put C_1 into V_j and the delivery sequence is C_2, C_1, and C_3. Delete C_1 from V_i.
The swap procedure is to randomly select two routes: V_i and V_j. Then randomly select a customer C_1 in V_i and a customer C_2 in V_j that are nearest to C_1. Exchange C_1 and C_2.
The new-route generation procedure is to randomly select a route V_i. Then randomly select a customer C_1 in V_i and put it into a new generated route V_j. Delete C_1 from V_i.

Step 20. Are the total demands of customers in V_i and V_j less than or equal to MaxSup and the order quantities during each production run in V_i and V_j less than or equal to vehicle capacity? If yes, go to step 21. Otherwise, go to step 19.

Step 21. Is the candidate move in the tabu list (The length of tabu list is set equal to 7 [8].)? If yes, go to step 22. Otherwise, (1) update $X^0 = X^1$, $SC(X^0) = SC(X^1)$, (2) update the tabu list at the routing and inventory stage, and (3) go to step 23.

Step 22. Is $SC(X^1) \leqslant SC(X^*)$? If yes, (1) update $X^* = X^1$, $SC(X^*) = SC(X^1)$, (2) update $X^0 = X^1$, $SC(X^0) = SC(X^1)$, (3) update the tabu list at the routing and inventory stage, and (4) go to step 23. Otherwise, go to step 19.

Step 23. Initialize the parameters for the simulated annealing search such as initial temperature (= 70), reheating factor r (= 0.9), stopping temperature (= 10).

Step 24. Generate a neighboring solution X^1 as the next candidate solution using insertion procedure, swap procedure, or new-route generation procedure randomly (please refer to step 19 for details of these three procedures).

Step 25. Are the total demands of customers in V_i and V_j less than or equal to MaxSup and the order quantities during each production run in V_i and V_j less than or equal to vehicle capacity? If yes, go to step 26. Otherwise, go to step 24.

Step 26. Is the move (from X^0 to X^1) in the tabu list? If yes, go to step 24. Otherwise, go to step 27.

Step 27. Is the neighboring solution accepted?
1. Let $\Delta SC = SC(X^1) - SC(X^0)$.
2. If $\Delta SC \leqslant 0$, then $X^0 = X^1$, $SC(X^0) = SC(X^1)$ and update the tabu list at the routing and inventory stage. And if $SC(X^1) \leqslant SC(X^*)$, then $X^* = X^1, SC(X^*) = SC(X^1)$.
3. If $\Delta SC > 0$, then $X^0 = X^1$, $SC(X^0) = SC(X^1)$ with probability $\exp(-\Delta SC/T)$ and update the tabu list at the routing and inventory stage.

Step 28. Should the procedure stop under the temperature T? If yes, go to step 29; otherwise, go to step 19.
Please refer to step 16 for details.

Step 29. $T = T \times r$.

Step 30. Is the stopping criterion ($T <$ stopping temperature) at the routing and inventory stage matched? If yes, go to step 31; otherwise, go to step 19.

Stopping criterion is matched?

Step 31. Is the count equal to max_count (= 5)? If yes, stops. Otherwise, (1) set count = count + 1 and (2) go to step 7.

3 Computation results and comparisons

In order to examine the computational effectiveness and efficiency of the proposed heuristic method (H_1), three methods are used to compare with the proposed method. The first method is a heuristic (H_2) proposed by [3] (please refer to Appendix B for details). The other two methods are simulated annealing search (H_3) and tabu search (H_4) (please refer to Appendix C for details). The heuristic methods are coded using Visual C++ programming language and the tests are carried out on a PC Pentium 1.4 GHz.

For evaluating the proposed heuristic H_1, the test problems are divided into two categories: size small and large. For small-sized problems with up to four candidate depots and eight customers, the solutions for H_1 H_2, H_3 and H_4 are compared to the optimal solution yielded by enumeration search. A set of 45 tests classified in nine different problem sizes (2 candidate depots × 4 customers, 2 candidate depots × 6 customers, 2 candidate depots × 8 customers, 3 candidate depots × 4 customers, 3 candidate depots × 6 customers, 3 candidate depots × 8 customers, 4 candidate depots × 4 customers, 4 candidate depots × 6 customers, 4 candidate depots × 8 customers) was designed to evaluate the performance of the heuristic solutions versus the optimal solutions. Each problem instance contained five tests. The detailed settings for each test problem are as follows [3]:

1. The demand for each customer is selected from a uniform distribution U[450, 600] for each month.
2. The demand during lead time for each customer is selected from a uniform distribution U[0, 10].
3. The location (x, y) of each customer and candidate depot is selected from a uniform distribution U[0, 100].
4. The vehicle capacity is 300.
5. The vehicle service capacity is 4000.
6. The fixed ordering cost is 20.
7. The vehicle dispatching cost is 25
8. The shortage cost is 2.
9. The holding cost is 0.5.
10. The distance cost is 1/unit distance.

Table 1 shows the average solution quality and average CPU times for H_1, H_2, H_3, H_4 and optimal solutions. It can be seen that the proposed heuristic solutions (H_1) are better than or equal to those of H_2, H_3, and H_4 and near optimal (or optimal) in different small-sized problems. The average CPU times are less than or equal to 16.5 s for H_1, H_2, H_3, and H_4. However, the maximal average CPU time for obtaining optimal solutions is 19, 371 s. The larger the problem size, the larger the computational time for obtaining optimal solutions. The heuristic methods are more efficient than the optimal procedure.

For larger-sized problems, the optimal solutions cannot be obtained in a reasonable time and there is no tight lower bound for this problem. The performance of the proposed heuristic is evaluated against the solutions of H_2, H_3 and H_4. A set of 45 tests classified in nine different problem sizes (10 candidate depots $\times$ 100 customers, 10 candidate depots $\times$ 150 customers, 10 candidate depots $\times$ 200 customers, 15 candidate depots $\times$ 100 customers, 15 candidate depots $\times$ 150 customers, 15 candidate depots $\times$ 200 customers, 20 candidate depots $\times$ 100 customers, 20 candidate depots $\times$ 150 customers, 20 can-

Table 1. Results for small-sized problems

Number of candidate depots	Number of customers	Optimal cost[1]	Optimal cpu[2]	H_1 cost[1]	H_1 cpu[2]	H_2 cost[1]	H_2 cpu[2]	H_3 cost[1]	H_3 cpu[2]	H_4 cost[1]	H_4 cpu[2]
2	4	2194.6	0.05	2194.6	0.05	2368	0.05	2194.6	0.05	2194.6	0.05
2	6	3229.8	2	3229.8	1	5189	0.05	3229.8	1	3229.8	1
2	8	3787.6	146.6	3805.6	15.5	6232.4	0.05	3956	15.2	3850.6	11.4
3	4	2178.2	0.05	2178.2	0.05	2302	0.05	2178.2	0.05	2178.2	0.05
3	6	3126	11.6	3126	2.9	5135.4	0.05	3126	2.9	3126	2.6
3	8	3750	2566.8	3799.4	16	6122.8	0.05	3921.2	15.3	3846	11.5
4	4	2092.2	0.05	2092.2	0.05	2289.6	0.05	2092.2	0.05	2092.2	0.05
4	6	3073.8	48.6	3192.6	8.1	5059.6	0.05	3273	7.3	3261.8	7
4	8	3745.6	19,371	3790	16.5	6016.2	0.05	3900	15.8	3835.6	12

[1] The average system cost of five test problems per instance.
[2] The average CPU time of five test problems per instance (s).

Table 2. Results for larger-sized problems

Number of candidate depots	Number of customers	H_1 cost[1]	H_1 cpu[2]	H_2 cost[1]	H_2 cpu[2]	H_3 cost[1]	H_3 cpu[2]	H_4 cost[1]	H_4 cpu[2]
10	100	44 552.6	40	68 958.6	5	51 970	40.6	45 618	44
10	150	64 692.8	67.5	93 816.4	7	68 150.4	67.6	65 931.4	79
10	200	84 231.8	141.4	122 073.8	8	86 587	146	86 605.4	134
15	100	43 499.4	72	68 132	8	48 631	70.8	44 712.6	60.8
15	150	62 634.6	90	90023.4	9	64 089.4	85.2	63 736.4	82
15	200	80 465.6	152.4	119 784.4	10.6	83 927.2	145	83 808.4	148.6
20	100	42 239.8	105.6	63 812.8	14.2	46 455.4	105	43 831	105
20	150	62 308.2	123.6	90 010.8	16.8	63 617.4	132.8	63 450.6	140.8
20	200	79 233.6	160.2	119 361	24	82 561.8	155	81 819.4	158.2

[1] The average system cost of five test problems per instance.
[2] The average CPU time of five test problems per instance (s).

Table 3. Results for problems with different levels for vehicle service capacity and vehicle capacity

Vehicle service capacity	Vehicle capacity	H_1 cost[1]	H_1 cpu[2]	H_2 cost[1]	H_2 cpu[2]	H_3 cost[1]	H_3 cpu[2]	H_4 cost[1]	H_4 cpu[2]
2000	100	175 239.8	161.6	225 250	20	188 165	159.8	181 136	160
2000	200	104 301	163.8	121 148	21	110 129.6	157.6	107 688.6	163.6
2000	300	79 275.6	166.2	87 767	19.8	89 578.8	157.8	81 827.4	160.6
3000	100	175 298.8	163	278 408	24.2	188 099.8	156	181 222.6	161
3000	200	104 245	166	151 410	23.6	110 291.6	158.6	107 626.6	162
3000	300	79 201.4	160.2	105 875	23	89 568.6	156.8	81 832.4	161
4000	100	175 216.8	163	322 896	25	188 013	159.6	181 018	162
4000	200	104 231.6	164.4	167 418	24.6	110 321	157.6	107 708.6	161
4000	300	79 233.6	160.2	119 361	24	89 561.8	155	81 819.4	158.2

[1] The average system cost of five test problems per instance.
[2] The average CPU time of five test problems per instance (s).

didate depots × 200 customers) was designed to evaluate the performance of the heuristic solutions. Each problem instance contained five tests. The detailed settings are the same as those in small-sized problems. Table 2 shows the average solution quality and average CPU times for H_1, H_2, H_3, and H_4 for larger-sized problems. It is found that H_1 is better than H_2, H_3, and H_4 in terms of average system cost in all problems. When the number of candidate depots increases, the average system costs for H_1, H_2, H_3, and H_4 decrease because more choices for depot locations are available. When the number of customers increases, the average system costs for H_1, H_2, H_3, and H_4 increases because the transportation and inventory cost increases to meet the demands of increased customers. As for the computational time, it increases for these four methods when the problem size becomes large. Although the CPU times for H_1 is higher than those for H_2, the average system costs for H_1 are much less than those for H_2. In addition, the CPU times for H_1 are acceptable. Hence it is worthwhile searching for better solutions based on H_1.

Besides the problem size, vehicle service capacity and vehicle capacity are the other parameters that might affect the method performance [3, 7, 11]. For evaluating the performance of the heuristic methods under these two parameters mentioned above, a set of 45 tests classified in nine different problems was designed to evaluate the performance of the heuristic methods. Each problem instance contained five tests. The detailed settings for parameters are as follows:

1. The vehicle service capacity is set at 2000, 3000 and 4000.
2. The vehicle capacity is set at 100, 200 and 300.
3. The number of candidate depots is 20.
4. The number of customers is 200.

The other settings for cost parameters and system parameters are the same as those mentioned in small-sized problems. Table 3 shows the average solution quality and average CPU times for H_1, H_2, H_3, and H_4 with different levels for vehicle service capacity and vehicle capacity. It is found that H_1 is better than H_2, H_3, and H_4 in terms of average system cost. When the vehicle service capacity increases, the average system costs for H_1, H_3, and H_4 are almost the same and they increase for H_2. When the vehicle capacity increases, the average system costs for H_1, H_2, H_3, and H_4 decrease. As for the computational time, these four methods are not affected by the parameters of vehicle service capacity and vehicle capacity.

4 Conclusions

In this paper, we have developed an effective heuristic method for the combined location routing and inventory problem, which still remains as computationally intractable. The proposed heuristic method is better than those heuristic methods searching for local optima and pure global search heuristic methods in terms of average system cost. Though the computation time for the proposed method is longer than the local optima search method, it is still an acceptable and promising method.

Two related research directions are as follows: (1) develops a method for solving the CLRIP taking other constraints into considerations such as multiple vehicle fleet types, time window for customers' demand, etc. and (2) develops a multiple-objective model for the three parts: manufacturers, distributed service suppliers, and retailers.

Acknowledgement This work was partially supported by the National Science Council, Taiwan, R.O.C., under grant NSC92-2213-E-020-009.

Appendices

Appendix A

Notations

M:	Number of candidate depots
N:	Number of customers
K:	Number of vehicles (or routes)
b:	Vehicle capacity
MaxSup:	Vehicle service capacity
FC_j:	Cost of establishing depot j
c:	Cost of dispatching vehicles
cm:	Traveling cost / unit distance
h^+:	Holding cost / unit time/ unit product
hs:	Shortage cost / unit product
A:	Ordering cost / each order
h:	Index of depots or customers ($1 \leqslant h \leqslant N+M$)
g:	Index of depots or customers ($1 \leqslant g \leqslant N+M$)
i:	Index of customers ($1 \leqslant i \leqslant N$)
j:	Index of depots ($N+1 \leqslant i \leqslant N+M$)
k:	Index of vehicles or routes ($1 \leqslant k \leqslant K$)
V_k:	Set for route k with an open depot ($1 \leqslant k \leqslant K$)
Dis_{kgh}:	Total distance for route k.
Q_{kgh}:	Number of units produced for route k during each production run
UL_{kgh}:	Average demand for route k during lead time
D_{kgh}:	Total demand for route k
R_{kgh}:	Order-up-to level for replenishment of route k
$B(R_{kgh})$:	Expected shortage number for route k during each production run
Y_{ij}:	1, if customer j is allocated to depot i; 0 otherwise
Z_j:	1, if depot j is established; 0 otherwise
X_{kgh}:	1, if point g immediately proceeds point h on route k; 0 otherwise

The model formulation is as follows:

$$\text{Minimize} \sum_{j=N+1}^{N+M} FC_j \times Z_j + \sum_{k=1}^{K} \sum_{g=1}^{N+M} \sum_{h=1}^{N+M} \left((c + cm \times Dis_{kgh}) \times \frac{D_{kgh}}{Q_{kgh}} + \left(\frac{Q_{kgh}}{2} + R_{kgh} - UL_{kgh} \right) \times h^+ + \frac{D_{kgh}}{Q_{kgh}} \times A + hs \times B(R_{kgh}) \times \frac{D_{kgh}}{Q_{kgh}} \right) \times X_{kgh} \quad \text{s.t.}$$

$$Q_{kgh} \leqslant b \quad \text{(A.1)}$$

$$D_{kgh} \leqslant MaxSup \quad \text{(A.2)}$$

$$\sum_{k=1}^{K} \sum_{h=1}^{N+M} X_{ikh} = 1, \quad i = 1, \ldots, N \quad \text{(A.3)}$$

$$\sum_{g \in v} \sum_{h \in v} \sum_{K=1}^{k} X_{ghk} \geqslant 1, \quad \forall (v, \bar{v}) \quad \text{(A.4)}$$

$$\sum_{g=1}^{N+M} X_{hgk} - \sum_{g=1}^{N+M} X_{ghk} = 0, \quad k = 1, \ldots, K, \quad h = 1, \ldots, N+M \quad \text{(A.5)}$$

$$\sum_{j=N+1}^{N+M} \sum_{i=1}^{N} X_{ijk} \leqslant 1, \quad k = 1, \ldots, K \quad \text{(A.6)}$$

$$\sum_{h=1}^{N+M} X_{ihk} + \sum_{h=1}^{N+M} X_{jhk} - Y_{ij} \leqslant 1, \quad i = 1, \ldots, N, \quad j = N+1, \ldots, N+M, \quad K = 1, \ldots, K \quad \text{(A.7)}$$

$$X_{kgh} = 0, 1, \quad g = 1, \ldots, N+M, \quad h = 1, \ldots, N+M, \quad k = 1, \ldots, K \quad \text{(A.8)}$$

$$Z_j = 0, 1, \quad j = N+1, \ldots, N+M \quad \text{(A.9)}$$

$$Y_{ij} = 0, 1, \quad i = 1, \ldots, N, \quad j = N+1, \ldots, N+M \quad \text{(A.10)}$$

In the above formulation, the objective function is to minimize the sum of depot establishing cost, transportation cost, and inventory cost. Constraint Eq. A.1 states the amount of each delivery to customers must be less than or equal to vehicle capacity. Constraint Eq. A.2 insures the total demand for route k is less than or equal to vehicle service capacity. Constraint Eq. A.3 states each customer only appears in one route. Constraint Eq. A.4 insures each route begins and ends at the same depot. Constraint Eq. A.5 insures that every point entered by the vehicle should be the same point the vehicle leaves. Constraint Eq. A.6 insures a route cannot be served by multiple depots. Constraint Eq. A.7 states a customer can be allocated to a depot only if there is a route passing by that customer. Constraints Eqs. A.8 –A.10 insure the integrality of decision variables.

Appendix B

Step 1. (1) Set $k = 1$, $r = 1$, MaxSup = vehicle service capacity, max_swap = 0. (2) Put all customers into a set F. (3) Put all depots into a set E.

Step 2. (1) Randomly select a customer from F. (2) Put this customer into the set V_k. (3) Delete the customer from F.

Step3. Select a customer, W, from F with the minimal marginal cost C_s as the next candidate customer.

Step 4. Is the total demand of customers in V_k and the candidate customer less than or equal to MaxSup? If yes, (1) put the candidate customer into V_k, (2) delete the candidate customer from F, (3) go to step 5. Otherwise, (1) set k = k+1, (2) put the candidate customer into V_k, (3) delete the candidate customer from F, (4) go to step 5.

Step 5. Is F empty? If yes, go to step 6. Otherwise, go to step 3.

Step 6. Compute the centroid of V_t for $1 \leqslant t \leqslant k$.

Step 7. (1) Select a depot from E with the shortest path to the centroid of V_r. (2) Put the depot into V_r. (3) Delete the depot from E. (4) Set $r = r + 1$.

Step 8. Is r greater than k? If yes, go to step 9. Otherwise, go to step 7.

Step 9. (1) Compute the total system cost SC. (2) Set the initial solution SC and V_t as the temporary best solution (Let SC' = SC and $U_t = V_t$ for $1 \leqslant t \leqslant k$).

Phase 2: improving the initial solution

Step 10. (1) Close a depot with the minimal system cost SC. (2) Set NOD = NOD − 1 (The initial value of NOD is equal to the number of routes k.).

Step 11. Is SC less than SC'? If yes, set SC' = SC, $U_t = V_t$ for $1 \leqslant t \leqslant k$ and go to step 12. Otherwise, go to step 12.

Step 12. Is NOD equal to 1? If yes, go to step 13. Otherwise, go to step 10.

Step 13. (1) Set $V_t = U_t$ for $1 \leqslant t \leqslant k$. (2) Randomly select a closed depot and an open depot from the best solution and exchange each other (the selected open depot is substituted by the selected closed depot in V_t). (3) Compute the total system cost SC.

Step 14. Is SC less than SC'? If yes, set SC' = SC, $U_t = V_t$ for $1 \leqslant t \leqslant k$ and go to step 15. Otherwise, set max_swap = max_swap + 1 and go to step 15.

Step 15. Is max_swap greater than a default value (usually the default value is set equal to half of the number of candidate depots)? If yes, go to step 16. Otherwise, go to step 13.

Step 16. The best solution (SC' and U_t for $1 \leqslant t \leqslant k$) is obtained and stops.

Appendix C

The detailed procedure for SA is as follows:

Step 1. Obtaining the initial solution (please refer to steps 1–6 of the proposed heuristic method in Sect. 2).

Step 2. Improving solution quality at the depot location-allocation stage (please refer to steps 11–18 of the proposed heuristic method in Sect. 2).

Step 3. Improving solution quality at the routing and inventory stage (please refer to steps 23–30 of the proposed heuristic method in Sect. 2).

Step 4. Is the count equal to max_count (= 5)? If yes, stops. Otherwise, (1) set count = count + 1 and (2) go to step 2.

The detailed procedure for TS is as follows:

Step 1. Obtaining the initial solution (please refer to step 1–6 of the proposed heuristic method in Sect. 2).

Step 2. Improving solution quality at the depot location-allocation stage (please refer to steps 7–10 of the proposed heuristic method in Sect. 2).

Step 3. If $SC(X^1) > SC(X^*)$, no_improving_depot = no_improving_depot + 1.

Step 4. Is the number of non-improving moves no_improving_depot at the depot location-allocation stage less than max_depot (= 150)? If yes, go to step 2. Otherwise, go to step 5.

Step 5. Improving solution quality at the routing and inventory stage (please refer to steps 19–22 of the proposed heuristic method in secref2).

Step 6. If $SC(X^1) > SC(X^*)$, no_improving_routing = no_improving_routing + 1.

Step 7. Is the number of non-improving moves no_improving_routing at the routing and inventory stage less than max_routing (= 2000)? If yes, go to step 5. Otherwise, go to step 8.

Step 8. Is count equal to max_count (= 5)? If yes, stops. Otherwise, (1) set count = count + 1 and (2) go to step 2.

References

1. Viswanathan S, Mathur K (1997) Integrated routing and inventory decisions in one-warehouse multiretailer multiproduct distribution system. Manage Sci 43:294–312
2. Hall RH (1987) Consolidation strategy: inventory, vehicle and terminals. J Bus Logist 8:57–73
3. Liu SC, Lee SB (2003) A two-phase heuristic method for the multi-depot location routing problem taking inventory control decision into considerations. Int J Adv Manuf Technol, in press
4. Wu TH, Low C, Bai JW (2002) Heuristic solutions to multi-depot location-routing problems. Comput Oper Res 29:1393–1415
5. Tan KC, Lee LH, Ou K (2001) Artificial intelligence heuristics in solving vehicle routing problems with time window constraints. Eng Appl Artif Intell 14:825–837
6. Renaud J, Laporte G, Boctor FF (1996) A tabu search heuristic for the multi-depot vehicle routing problem. Comput Oper Res 23: 229–235
7. Tuzun D, Burke LI (1999) A two-phase tabu search approach to the location routing problem. Eur J Oper Res 166: 87–99
8. Glover B (1977) A statistical approach to the TSP. Networks 7:205–229
9. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:167–175
10. Skiscim CC, Golden BL (1983) Optimization by simulated annealing: a preliminary computational study for the TSP. Winter Simulation Conference, 1983
11. Tan KC, Lee LH, Zhu QL, Ou K (2001) Heuristics methods for vehicle routing problems with time windows. Artif Intell Eng 15:281–295