2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Vehicle Routing Problems with Time Windows Using Simulated Annealing

S.-W. Lin, K.-C. Ying, Z.-J. Lee and H.-S. Chen

Abstract—**In recent years, supply chain management is paid attention to every large enterprise. Enterprise reduces their transporting and distributing cost through the manner of subcontractors and shared transportation. The customer request for sending time of the goods is getting more strictly; it can be solved in the vehicle routing problem with time windows (VRPTW). Because the constraints of VRPTW include the length of each route, loading capacity of vehicle and the available time window for each customer, it is more complex than travel salesperson problem and vehicle routing problem (VRP).**

This research applied the simulated annealing (SA) combined with local search for solving the VRPTW. The developed approach can escape from the local optimal traps, and the use of exchange and insertion local search can find out the (near) optimal solution quickly and efficiently. The Solomon's benchmark instances are used for verifying the developed approach. All problems have 100 customers, a delivery depot, constraints of loading capacity and time window. The developed approach finds all the best results in the C set, and find out 4 solutions which are equal to the best solutions found so far in R set and RC set at reasonable computational time. The developed approach finds the average number of vehicles and route costs in most classes are better than or equal to those of previous researches. Therefore, the developed approach can be used to solve the VRPTW effectively.

I. Introduction

As the living standards have increased, customers pay increasing attention to the accuracy of product delivery times. As a result, the planning of delivery routes must consider customers' acceptable service time window. Therefore, how to reduce the cost while delivering on time, namely, vehicle routing problems with time windows (VRPTW), has become an important issue in supply chain management.

The objective of VRPTW is to design the shortest path for minimum traveling costs and number of vehicles without violating the constraints of time windows and loading capacity of vehicle. A vehicle starts from one depot to deliver goods to a set of scattered customers. Each vehicle's time of delivery to customers must within the customer's time window. If the arrival time is earlier than the time window, the vehicle must wait to deliver the goods until the beginning of customer's time window. Total deadweight of each vehicle cannot exceed the constraint of the vehicle capacity, and the vehicle must get back to the depot within the time that the depot stipulates finally.

VRPTW is a NP-hard problem because many factors need to be taken into consideration and there are numerous possibilities of permutation and combinations [1]. Many researches proposed exact methods and heuristics to solve this type of problem. Kolen et al. [2] developed a branch and bound approach to solve the VRPTW. Desrochers et al. [3] proposed a column generation approach that solved the Solomon's benchmark instances. Fisher et al. [4] proposed a K-tree relaxation approach to solve two of Solomon's benchmark instances. Exact methods can guarantee the optimality, but is requires considerable computer resources in terms of both computational time and memory.

Clarke and Wright [5] were the first to propose the use of a savings algorithm to construct a feasible solution of vehicle routing problems. Miller and Gillett [6] applied a sweep algorithm to build a feasible solution of vehicle routing problems, but the quality of the solution of the sweep algorithm is not stable. Solomon [7] developed sequential insertion heuristics to construct a feasible solution. These routes construction methods can obtain an initial feasible solution quickly but the quality of solution may not be satisfied, especially for large size problem.

Local search approach can iteratively modify the current solution from neighboring solutions. A neighborhood comprises the set of solutions that can be researched from the current one by swapping a subset of r points (customer) in traveling sequence between solutions. An r-exchange [8]-[10] is implemented only if it leads to an improved feasible solution. It can be performed within or between routes. The process terminates when an r-optimal solution is found, that is, one that cannot be improved. Push forward insertion heuristic [7] improved the current solution by applying the insertion method to the routes of VRPTW. Local searches can obtain a better feasible solution than initial feasible solution, but the solution obtained is a local optimal.

Therefore, many researchers have adopted meta-heuristics to handle this type of NP-hard problems. Chiang and Russell [11] proposed partial routes constructed and added a tabu list which is used to avoid cycling in the simulated annealing

Shih-Wei Lin is an assistant professor with the department of information management, Huafan University, Taiwan (e-mail:swlin@cc.hfu.edu.tw; phone: 886-2-26632102#4373; Fax: 886-2-26632102#4353).

Kuo-Ching Ying is an associate professor and the chairman with the department of industrial engineering and management information, Huafan University, Taiwan.

Zne-Jung Lee is an assistant professor with the department of information management, Huafan University, Taiwan.

Hong-Sheng Chen is a graduate student with the department of information management, Huafan University, Taiwan

(SA). Osman [12] used SA and tabu search (TS) for the VRPTW, the result showed that interchange methods guided by SA and TS produce good solutions. Haibing and Andrew [13] proposed a meta-heuristic based on annealing-like restarts to diversify and enhance local searches for solving the VRPTW. Taillard et al. [14] used CROSS exchange, Or-opt exchange and GENIUS as the core of TS, Potvin et al. [15] used 2-opt* and Or-opt as the core of TS, Gehring and Homberger [16], [17] applied a similar approach with parallel TS implementation to solve the VRPTW. Blanton and Wainwright [18] were the first to apply a GA and hybridized a GA with a greedy heuristic in VRPTW. Bräysy et al. [19] hybridize a GA with an evolutionary algorithm consisting of various route construction and improvement heuristics. Gambardella et al. [20] applied an Ant Colony Optimization (ACO) based approach useful to solve vehicle routing problems with time windows. All these meta-heuristics can help to obtain better solutions for VRPTW by exploring much larger solution space.

II. Problem Definition

Potvin and Bengio [21] defined VRPTW as follows. Given one delivery depot, same type of vehicles, and known locations, demands and time windows of customers, customers' demands cannot exceed the load capacity of the vehicle, each customer can only be served by a single vehicle, and vehicles must return to the depot within the time limit of depot. The main objective is to have minimum vehicle numbers and the shortest total route distance without violating the constraints of vehicles' loading capacity and time windows. Under the constraint of time windows, vehicle routing problems need to consider three factors: routing, loading and scheduling. Moreover, a delivery depot also has the constraint of a time window, causing the constraint of route length when vehicles deliver. Therefore, the complexity of VRPTW is higher than the complexities of traveling salesman problem (TSP) and vehicle routing problem.

VRPTW can be stated and solved by mathematical programming models [22] as shown in follows.

Decision Variables:

t_i arrival time at customer i;

w_i waiting time at customer i;

$x_{ijk}=1$ if there vehicle k travels from customer i to customer j, and 0 otherwise. ($i \neq j$; i, j=0, 1, ..., N).

Parameters:

V total number of vehicles,

N total number of customers,

c_i customer i (i=1, 2, ..., N),

c_0 delivery depot,

c_{ij} traveling distance between customer i to customer j,

t_{ij} travel time between customer i and customer j,

m_i demand of customer i,

q_v capacity of vehicle v,

e_i earliest arrival time at customer i;

l_i latest arrival time at customer i;

f_i service time at customer i;

r_v maximum route time allowed for vehicle v;

Minimize

$$\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{v=1}^{V} c_{ij}x_{ijv} \tag{2.1}$$

Subject to

$$\sum_{v=1}^{V}\sum_{j=1}^{N} x_{ijv} \le V \qquad \text{for } i=0, \tag{2.2}$$

$$\sum_{j=1}^{N} x_{ijv} = \sum_{j=1}^{N} x_{jiv} \le 1 \qquad \text{for } i=0 \text{ and } v\in\{1, \ldots, V\}, \tag{2.3}$$

$$\sum_{v=1}^{V}\sum_{j=0}^{N} x_{ijv} = 1 \qquad \text{for } i\in\{1, \ldots, N\}, \tag{2.4}$$

$$\sum_{v=1}^{V}\sum_{i=0}^{N} x_{ijv} = 1 \qquad \text{for } j\in\{1, \ldots, N\}, \tag{2.5}$$

$$\sum_{i=0}^{N} m_i \sum_{j=0}^{N} x_{ijv} \le q_v \qquad \text{for } v\in\{1, \ldots, V\}, \tag{2.6}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} x_{ijv}(t_{ij}+f_i+w_i) \le r_v \qquad \text{for } v\in\{1, \ldots, V\}, \tag{2.7}$$

$$t_0 = w_0 = f_0 = 0, \tag{2.8}$$

$$\sum_{v=1}^{V}\sum_{i=0}^{N} x_{ijv}(t_i+t_{ij}+f_i+w_i) = t_j \qquad \text{for } j\in\{1, \ldots, N\}, \tag{2.9}$$

$$e_i \le (t_i+w_i) \le l_i \qquad \text{for } i\in\{0, \ldots, N\}, \tag{2.10}$$

Formula (2.1) is the objective function of the problem. The first set of constraints (2.2) specifies that there are at most V routes going out of the depot. The second set of constraint (2.3) makes sure every route starts and ends at the delivery depot. The third set of constraints (2.4) and the forth set of constraint (2.5) restrict the assignment of each customer to exact one vehicle route. The fifth set of constraints (2.6) ensures the loading capacity of vehicle will not be violated. The sixth set of constraints (2.7) is the maximum travel time constraint. Other sets of constraints (2.8)–(2.10) guarantee schedule feasibility with respect to time windows.

The scale of the problem depends on the number of constraints. When N is small, traditional mathematical programming approaches can be used to obtain the real optimal solution of VRPTW; however, when N is large, it is not possible to do that. Therefore, researchers have developed various algorithms that can finish performing within polynomial time to find the problem's initial feasible solution and then apply the meta-heuristic approach to obtain (near) global optimum solution.

This study uses SA combined with local search to solve VRPTW. Its principle is: (1) in local search, starting from one solution and search the neighborhood of this solution through exchange and insertion, a better solution may be sought. (2) SA is a global search meta-heuristic, it can avoid falling into local optimum while solving, which has the opportunity of jumping out of the local optimum, and further seek a (near) global optimum solution. Local search can find a better solution than current one; however, this better solution may be a local optimum solution. SA can jump out of local optimum. Therefore, SA combined with the local search approach can more effectively find the (near) global optimum solution.

III. Developed Approach

SA was originated by Metropolis et al. [23]. Kirkpatrick et al. [24] are credited for its widespread applications. SA was developed from the so-called "statistical mechanics" idea. Annealing is the process through which slow cooling of metal produces good, low energy state crystallization, whereas fast cooling produces poor crystallization. The optimization procedure of simulated annealing reaching a (near) global minimum mimics the crystallization cooling procedure.

Generally, suddenly reducing high temperature to very low (quenching) cannot obtain this crystalline state. In contrast, the material must be slowly cooled from high temperature (annealing) to obtain crystalline state. During the annealing process, every temperature must be kept long enough time to allow the crystal to have sufficient time to find its minimum energy state. The local search continuously seeks the solution better than the current one during the searching process. If search procedure only accepts the solution whose objective function value is smaller than current one during the iterative process, which is just like the quenching process, will be trapped in local optimum easily. On the other hand, the iterative process of SA permits "uphill", that is, making the feasible solution of the objective function value slightly upward (worse) in order to strip out the local optimum and find the global optimum.

When applying the developed approach to solve the VRPTW, we need to decide the solution representation of vehicle routes. The solution representation uses 0 to represent the delivery depot. The first vehicle must start from the depot, and then visit the customer sequentially according to the number in solution representation. According to the constraint of time window, each vehicle arrives at each customer must within customer's time window. A vehicle can arrive before the starting of the time window but still needs to wait until the allowable time of delivery; otherwise, it will violate the time window constraint. The vehicle must return to the depot within the time window of depot; other vehicles then leave in order, and the process is iterated until each customer is routed, but one customer can only be served by one vehicle. Given that a solution to a VRPTW is made of multiple routes, the path representation is extend and contains multiple copies of the depot, with each copy acting as a separator between two routes. For example, a solution representation as shown in Figure 1 would correspond to a VRPTW solution made of four routes. The first route contains customers 12, 17, 15, 10, 9, 16 and 5, the second route contains customers 11, 2, 20, 3 and 1, the third route contains customers 8, 7, 14 and 6, the forth route contains customers 18, 13, 19 and 4.

12,17,15,10,9,16,5,0,11,2,20,3,1,0,8,7,14,6,0,18,13,19,4

Fig. 1. An example of solution representation.

The above four routes can be displayed graphically as shown in Figure 2. Each route starts from depot, visiting customers and ends at depot.

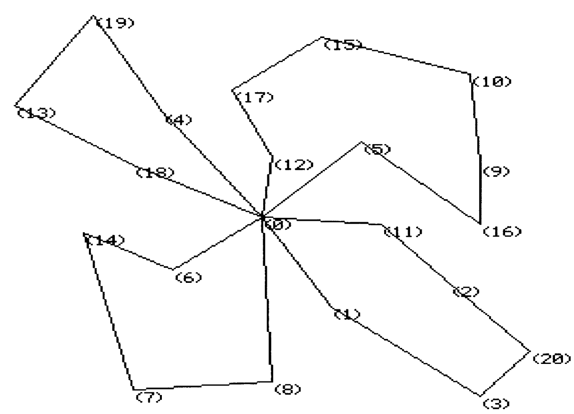

Fig.2. A graph representation of VRPTW solution.

This study used sequential insertion heuristic [7] to set out the initial solution of SA. Therefore, the initial solution is a feasible solution, and the routes created by the initial feasible solution can be used in the local search algorithm. In the local search algorithm, we adopted 2-opt exchange and insertion methods. When SA was looking for the next solution, there was a certain probability of choosing 2-opt exchange or insertion method to improve the current solution. In addition, there was a certain probability for exchanging or inserting with nearby 30 points in order to find to best solution. If new solution generated by 2-opt exchange or insertion violates the constraints of time windows and loading capacity, the penalties will be added to the objective function value, which make the solution have the worse objective function value. Sometimes accepting an infeasible solution may help to jump out the local optimal; therefore, infeasible solution can be accepted temporarily. Let Cost(S) be the objective function of solution S. Cost(S) can be calculated as $C_1N + C_2\mathrm{P}(S) + C_3\mathrm{Dist}(S)$, where N is the number of vehicles used, P(S) measure the degree of violation of constraint, Dist(S) is traveling distance of S, and the penalty weight factors $C_1 >> C_2 >> C_3$. When the SA process is finished, the minimum number of vehicle used and the minimum traveling distance can be output, if the final solution is feasible. During the local search process, if the solution obtained is better than the initial solution, then this solution obtained is regarded as the next solution. If the solution obtained is not better than the initial solution, probability determined by Metropolis criterion is used to decide whether this solution is going to replace the current solution or not. During the SA process, the best solution found so far will be recorded. After the SA process is finished, the best solution found so far is the solution we obtained.

The mechanism of 2-opt exchange works as follows. Randomly choose one point (customer) from one route, and one point from another route, and then exchange the two points, as shown in Figure 3.

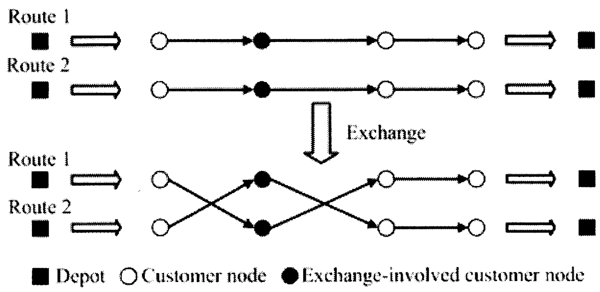

Fig. 3. 2-opt exchange sketch map.

The mechanism of insertion operates as follows. Randomly pick and delete one point from one route and insert this point into a random position of another route, as shown in Figure 4.

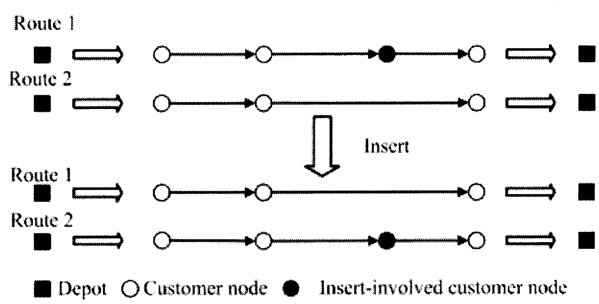

Fig. 4. Insertion sketch map

If two chosen points are in the same route, the 2-opt exchange and insertion can be done in the same way.

The SA procedure of this research is described as follows. At the beginning, the nearest algorithm is used to obtain S_0 as the initial solution, regarded as an energy state of the metallic physic system. Let $E(S_0)$ be the objective function value for $E(S_0)$; $E(S_0)$ can be considered as the energy value of physic system. Control parameter T (analog of temperature) represents the annealed temperature. T_0 denotes initial temperature. When SA begins to run, T is set to be the same as T_0. In each temperature T, obtain the new configuration state (new solution) S_1 via local search. Calculate the difference $\triangle E= E(S_1)-E(S_0)$, between two objective function values. If $\triangle E$ is smaller than 0, i.e., $\triangle E<0$, which means the new configuration state S_1 is better than S_0, the system will then accept S_1 and replace S_0; the system will evolve to a new configuration state. If the difference $\triangle E$ is larger than zero, that is to say, S_1 is worse than S_0, The Boltzmann probability distribution will be used as acceptance criteria $\exp(-\triangle E/T)$ to judge accepting S_1 or not. We calculated the probability of $\exp(-\triangle E/T)$ and randomly selected, from uniform distribution U(0, 1), one real number r to compare. If $\exp(-\triangle E/T)>r$, the new configuration state is accepted as the new system configuration state. If $\exp(-\triangle E/T)<r$, the new configuration state is removed and the original system state is kept. After running C_{iter} times at a certain temperature, cool the temperature according to cooling schedule $T_{n+1}=\alpha T_n$, where α is cooling rate; it is between 0 and 1. After iterating the above loop until the temperature drop to stopping temperature T_F, SA algorithm process finishes.

IV. COMPUTATIONAL RESULT

The developed approach of this study uses C programming language, computers equipped with Pentium IV 3.0G MHz CPU, 512 MB memory, and Window XP operating system.

In order to verify the effectiveness of the developed approach, the VRPTW benchmark problem instances provided by Solomon [7] are used as the examples, and the results calculated from the developed approach are compared against the results of other approaches. In the benchmark problem instances, all problems are assumed to have one delivery depot and vehicles have the same loading capacity, each problem contains 100 customers. Each customer has the earliest and latest allowable service time (time window). Each vehicle has a constant loading capacity. Time and distance can be converted in equal units, and the amount of each customer's demand is known. The VRPTW benchmark problem instances have six sets: C1, C2, R1, R2, RC1, and RC2. Among them, problems in set C have clustered customers whose time windows were generated based on a known solution. Problems in set R have customers location generated uniformly randomly over a square. Problem in set RC have a combination of randomly placed and clustered customers. Problem sets of type 1 have narrower time windows and smaller vehicle loading capacity. Problem set of type 2 have larger time windows and large vehicle loading capacity. Therefore, the solutions of type 2 problems have the fewer routes and significantly more customers per route.

Through the initial experiments, parameters of developed approach are set as follows. $T_0=100$, $C_{\text{iter}}=10000$, $\alpha=0.99$ and $T_F=0.1$. Each problem is solved 15 times, and the best one among 15 runs is taken as the objective function values obtained. The computational time of one run for one problem is range from 220 to 240 seconds using Pentium IV 3.0 GHz personal computer.

After carrying out the developed approach, problems in set C all found known best solutions, and problems in set R and RC obtained four solutions equal to the best solution found so far, respectively, as shown in Table 1. Comparing to the results of previous studies (Liu & Shen [25]; Gambardella et al. [20]; Gehring & Homberger [26]; Rousseau et al. [27]; Berger et al. [28]; Li et al. [29]; Tan et al. [30]; Zhu [31]; Beatrice et al. [32]), the developed approach found more solutions which are close to the best solution found so far, in views of the number of vehicle and route cost, in problems set R1, R2, RC1 and RC2. This result shows that the developed approach can effectively solve the vehicle routing problems with time window.

Table 1 shows the best results obtained from the proposed approach and the best published results obtained so far by approaches proposed in BVH [33], BBB [34], CC [35], CLM [36], GTA [20], HG [37], H [38], IKMUY [39], LLH [29], M [40], RT [41], RGP [26], SSSD [42], S97 [43] , S98 [44], and TBGGP [14]. In this table, NV means number of vehicles used, TD represents travel distance, Ref means the reference which obtained the corresponding best result.

TABLE 1.
OUR BEST SOLUTION AND BEST PUBLISHED.

Problem	Best published (NV / TD/ Ref)	Our best solution (NV / TD)
C101	10/828.94/ RT	**10/828.94**
C102	10/828.94/ RT	**10/828.94**
C103	10/828.06/ RT	**10/828.06**
C104	10/824.78/ RT	**10/824.78**
C105	10/828.94/ RT	**10/828.94**
C106	10/828.94/ RT	**10/828.94**
C107	10/828.94/ RT	**10/828.94**
C108	10/828.94/ RT	**10/828.94**
C109	10/828.94/ RT	**10/828.94**
C201	3/591.56/ RT	**3/591.56**
C202	3/591.56/ RT	**3/591.56**
C203	3/591.17/ RT	**3/591.17**
C204	3/590.60/ RT	**3/590.60**
C205	3/588.88/ RT	**3/588.88**
C206	3/588.49/ RT	**3/588.49**
C207	3/588.29/ RT	**3/588.29**
C208	3/588.32/ RT	**3/588.32**
R101	19/1645.79/H	19/1657.93
R102	17/1486.12/RT	**17/1486.12**
R103	13/1292.68/LH	14/1213.62
R104	9/1007.24/M	10/993.25
R105	14/1377.11/RT	**14/1377.11**
R106	12/1251.98/M	12/1257.18
R107	10/1104.66/S97	11/1076.78
R108	9/960.88/BBB	10/951.57
R109	11/1194.73/HG	12/1154.55
R110	10/1118.59/M	11/1087.53
R111	10/1096.72/RP	11/1066.27
R112	9/982.14/GTA	10/958.03
R201	4/1252.37/HG	4/1253.26
R202	3/1191.70/RGP	4/1085.70
R203	3/939.54/M	3/946.27
R204	2/825.52/BVH	3/749.63
R205	3/994.42/RGP	**3/994.42**
R206	3/906.14/SSSD	**3/906.14**
R207	2/893.33/BVH	3/814.74
R208	2/726.75/M	2/727.69
R209	3/909.16/H	3/915.64
R210	3/939.34/M	3/939.37
R211	2/892.71/BVH	3/783.75
RC101	14/1696.94/TBGP	15/1635.23
RC102	12/1554.75/TBGP	13/1477.54
RC103	11/1261.67/S98	11/1265.24
RC104	10/1135.48/CLM	10/1150.96
RC105	13/1629.44/BBB	14/1550.21
RC106	11/1424.73/BBB	12/1390.15
RC107	11/1230.48/S97	11/1233.21
RC108	10/1139.82/TBGP	10/1202.04
RC201	4/1406.91/M	4/1521.71
RC202	3/1367.09/CC	4/1183.10
RC203	3/1049.62/CC	3/1110.11
RC204	3/798.41/M	3/798.46
RC205	4/1297.19/M	4/1394.23
RC206	3/1146.32/H	3/1189.53
RC207	3/1061.14/BVH	3/1100.04
RC208	3/828.14/IKMUY	3/864.39

The average traveling distance and the average number of vehicles obtained are compared with those of other heuristics for the VRPTW as summarized in Table 2. It is noted that, our results match the best results on all problems in sets C1 and C2. And the average traveling distance and the average number of vehicles of most problems obtained are equal to or close to those of others in problem set R1, R2, RC1 and RC2.

TABLE 2.
COMPARISONS OF BEST AVERAGES ON BENCHMARK PROBLEMS.

Reference	C1	C2	R1	R2	RC1	RC2
Taillard et al. [14]	10.00	3.00	12.25	3.00	11.88	3.38
	828.45	590.30	1216.70	995.38	1367.51	1165.62
Homberger & Gehring [34]	10.00	3.00	11.92	2.73	11.63	3.25
	828.38	589.86	1228.06	969.95	1392.57	1144.43
Gehring & Homberger [16]	10.00	3.00	12.42	2.82	11.88	3.25
	829.00	590.00	1198.00	947.00	1356.00	1144.00
Liu & Shen [25]	10.00	3.00	12.17	2.82	11.88	3.25
	830.06	591.03	1249.57	1016.58	1412.87	1204.87
Gambardella et al. [20]	10.00	3.00	12.38	3.00	11.92	3.38
	828.38	591.85	1210.83	960.31	1388.13	1149.28
Tan et al. [30]	10.10	3.30	13.20	5.00	13.50	5.00
	861.00	619.00	1227.00	980.00	1427.00	1123.00
Gehring & Homberger [26]	10.00	3.00	12.00	2.73	11.50	3.25
	828.63	590.33	1217.57	961.20	1395.13	1139.37
Rousseau et al. [27]	10.00	3.00	12.08	3.00	11.63	3.38
	828.38	589.86	1210.21	941.08	1382.78	1105.22
Berger et al. [28]	10.00	3.00	12.17	2.73	11.75	3.25
	828.48	589.93	1230.22	1009.53	1397.63	1230.20
Li et al. [29]	10.00	3.00	12.08	2.91	11.75	3.25
	828.38	589.86	1215.14	953.43	1385.47	1142.48
Zhu [31]	10.00	3.00	12.80	3.00	13.00	3.70
	828.9	589.9	1242.70	1016.40	1412.00	1201.20
Berger & Barkaoui [34]	10.00	3.00	11.92	2.73	11.50	3.25
	828.48	589.93	1221.10	975.43	1389.89	1159.37
Beatrice et al. [32]	10.00	3.00	12.80	3.00	13.00	3.75
	828.47	590.60	1212.58	956.73	1379.86	1148.66
Our best solution	**10.00**	**3.00**	12.58	3.09	12.00	3.38
	828.38	**589.86**	1190.00	936.70	1363.07	1146.45

V. Conclusions and Future Research

This research used the sequential insertion heuristic to obtain the initial feasible solution of VRPTW and then utilized SA combined with local search to acquire a (near) global solution. When using the developed approach to solve the Solomon benchmark problem instances, problems in set C1 and C2 all known best solutions were found. The obtained solution of problems in set R1, R2, RC1 and RC2 are equal to or close to the solutions of previous studies. Thus, the developed approach can effectively find the (near) global optimum solution in Solomon's benchmark problem instances within a reasonable amount of time.

In the future, the local search of the developed approach can be applied to other similar problems, for example, capacitated VRP with pick-up and deliveries and time windows (CVRPPDTW), multiple depot VRP with time windows (MDVRPTW), periodic VRP with Time windows (PVRPTW), split delivery VRP with time windows (SDVRPTW), and so on. The essence of local search can also be used in other meta-heuristics, such as GA, TS and ACO, to solve similar problems.

REFERENCES

[1] M. Savelsbergh, "Local search for routing problems with time windows," Annals of Operations Research, vol. 4, 1985, pp. 285-305.

[2] A. W. J. Kolen, A. H. G.R. Kan, and H. W. J. M. Trienekens, "Vehicle routing with time windows," Operations Research, vol.35, pp.266-273.

[3] M. Desrochers, J. Desrosiers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," Operations Research, vol. 40, 1992, pp. 342-354.

[4] M. Fisher, K. Jörnsten, and O. Madsen, "Vehicle routing with time windows: Two optimization algorithms," Operations Research, vol. 45, 1997, pp. 488-492.

[5] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," Operations Research, vol. 12, pp. 568-581, 1964.

[6] B. Gillett and L. Miller, "A heuristic algorithm for the vehicle dispatch problem," Operations Research, vol. 22, pp. 340-349, 1974.

[7] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time windows constraints," Operations Research, vol. 35, 1987, pp. 254-265.

[8] R. A. Russell, "An effective heuristic for the m-tour traveling salesman problem with some side conditions," Operations Research, vol. 25, 1977, pp. 517-524.

[9] T. Cook and R. A Russell, "A simulation and statistical analysis of stochastic vehicle routing with timing constraints," Decision Science, vol. 9, 1978, pp. 673-687.

[10] E. Baker and J. Schaffer, "Computational experience with branch exchange heuristics for vehicle routing problems with time window constraints," American Journal of Mathematical and Management Sciences, vol. 6, 1986, pp. 261-300.

[11] W. Chiang, and R. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time windows," Annals of Operations Research, vol. 63, 1996, pp. 3-27.

[12] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems," Annals of Operations Research, vol. 42, 1993b, pp. 421-451.

[13] L. Haibing and L. Andrew, "Local search with annealing-like restarts to solve the VRPTW," European Journal of Operational Research, vol. 150, 2003, pp. 115-127.

[14] E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows," Transportation Science, vol. 31, 1997, pp.170-186.

[15] J. Y. Potvin, T. Kervahut, B. L. Garcia, and J. M. Rousseau, "The vehicle routing problem with time windows Part I: Tabu Search," Informs Journal on Computing, vol. 8, 1996, pp. 158-164.

[16] H. Gehring and J. Homberger, "A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows," In Proc. of EUROGEN99, University of Jyväskylä, Jyväskylä, Finland, pp. 57-64.

[17] H. Gehring and J. Homberger, "Parallelization of a two phase metaheuristic for routing problems with time windows," Asia-Pacific Journal of Operational Research, vol. 18, 2001, 35-47.

[18] J. Blanton and R. Wainwright, "Multiple vehicle routing with time and capacity constraints using genetic algorithms," In Proc. 5th International Conf. Genetic Algorithms, San Francisco, Morgan Kaufmann, 1993. p. 452-459.

[19] O. Bräysy, J. Berger, and M. Barkaoui, "A new hybrid evolutionary algorithm for the vehicle routing problem with time windows," Presented in Route 2000 Workshop, Skodsborg, Denmark, 2000.

[20] L. Gambardella, E. Taillard, and G. Agazzi, "MACSVRPTW: A multiple ant colony system for vehicle routing problems with time windows," IDSIA, Lugano, Switzerland, Tech. Rep. IDSIA-06-99, 1999.

[21] J. Y. Potvin and S. Bengio, "The vehicle routing problem with time windows Part II : Genetic Search," Informs Journal on Computing, vol. 8, 1996, pp.165-172.

[22] K. C. Tan, L.H. Lee, and K. Ou, "Artificial intelligence heuristics in solving vehicle routing problems with time window constraints, "Engineering Applications of Artificial Intelligence, vol. 14 , 2001, pp. 825–837.

[23] S. Kirkpatrick, C. D. Gelatt, and J. M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, 1983, pp. 671-680.

[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines, " Journal of Chemical Physics, vol.21, 1953, pp. 1087-1092.

[25] F.-H. Liu and S.-Y. Shen, "A route-neighborhood-based metaheuristic for vehicle routing problem with time windows," European Journal of Operational Research, vol.118, 1999, pp. 485–504.

[26] H. Gehring and J. Homberger, "Parallelization of a two phase metaheuristic for routing problems with time windows," Asia-Pacific Journal of Operational Research, vol. 18, 2001, 35-47.

[27] L.-M. Rousseau, M. Gendreau , and G. Pesant, "Using constraint-based operators to solve the vehicle routing problem with time windows," Journal of Heuristics, vol. 8, 2002, pp. 43-58.

[28] J. Berger, M. Barkaoui, and O. Bräysy, "A route-directed hybrid genetic approach for the vehicle routing problem with time windows," Information Systems and Operations Research, vol. 41, 2003, pp. 179-194.

[29] H. Li, A. Lim, and J. Huang, "Local search with annealing like restarts to solve the VRPTW," European Journal of Operational Research, vol.150, 2003, pp. 115-127.

[30] K. C. Tan, L. L. Hay, and O. Ke, "A hybrid genetic algorithm for solving vehicle routing problems with time window constraints," Asia-Pacific Journal of Operational Research, vol. 18, 2001, pp. 121–130.

[31] K. Q. Zhu, "A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows," in Proceedings of the 15th IEEE International Conference on Tools for Artificial Intelligence, 2003, pp. 176–183.

[32] O. Beatrice, J. R. Brian and F. Hanshar, "Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows," Springer Science Business Media, Inc. Manufactured in The Netherlands. Applied Intelligence 24, 2006, pp. 17–30.

[33] R. Bent and P. V. Hentenryck, "A two-stage hybrid local search for the vehicle routing problem with time windows," Department of Computer Science, Brown University, Tech. Rep. CS-01-06, 2001.

[34] J. Berger and M. Barkaoui, "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows," Computers & Operations Research, vol. 31, 2004, pp. 2037–2053.

[35] Z. J. Czech and P. Czarnas, "A parallel simulated annealing for the vehicle routing problem with time windows," in Proc. 10th Euromicro Workshop on Parallel, 2002, pp. 376-383.

[36] J. F. Cordeau, G. Larporte, and A. Mercier. "A unified tabu search heuristic for vehicle routing problems with time windows," Journal of the Operational Research Society, vol. 52, 2001, pp. 928-936.

[37] J. Homberger and H. Gehring, "Two evolutionary metaheuristics for the vehicle routing problem with time windows," INFOR, vol. 37, 1999, pp. 297-318.

[38] J. Homberger, "Verteilt-parallele metaheuristiken zur tourenplanung," Gaber, Wiesbaden, 2000.

[39] T. Ibaraki, S. Imahori, M. Kubo, T. Masuda, T. Uno, and M. Yagiura, "Effective local search algorithms for routing and scheduling problems with general time window constraints," Transportation Science, vol. 39, 2005, pp. 206-232.

[40] D. Mester, "An evolutionary strategies algorithm for large scale vehicle routing problem with capacitate and time windows restrictions," In Conf. Mathematical and Population Genetics, Univ. of Haifa, Israel, 2002.

[41] Y. Rochat and E.D. Taillard, "Probabilistic diversification and intensification in local search for vehicle routing," Journal of Heuristics, vol. 1, 1995, pp. 147-167.

[42] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck, "Record breaking optimization results using the ruin and recreate principle," Journal of Computational Physics, vol. 159, 2000, pp.139-171.

[43] P. Shaw, "A new local search algorithm providing high quality solutions to vehicle routing problems," Department of Computer Sciences, University of Strathclyde, Glasgow, Scottland, Tech. Rep, APES group, 1997.

[44] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in Proc. of the Fourth International Conf. Principles and Practice of Constraint Programming, New York, 1998, pp. 417-431.