Advances in Intelligent Transportation Systems
Vol. 1, No. 1, March 2012



# A Hybrid Approach for Vehicle Routing Problem with Time Windows

Shih-Wei Lin [1], So-Tsung Chou[2], Chou-Yuan Lee[3], and Zne-Jung Lee[4*]

[1]Department of Information Management, Chang Gung University, Taiwan
[2]Department of Information Management, Kang-Ning Junior College of Medical Care and Management, Taiwan
[3]Department of Information Management, Lan Yang Institute of Technology, Taiwan
[4*] Department of Information Management, Huafan University, No. 1, Huafan Rd. Shihding District, New Taipei County, 22301, Taiwan ( R.O.C.)

johnlee@hfu.edu.tw

**Abstract:** The vehicle routing problem with time windows (VRPTW) is an extension of the capacity constrained vehicle routing problem (VRP). 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 VRP. The VRPTW is NP-Complete and instances with 100 customers or more are very hard to solve optimally. This research applied a hybrid approach which takes the advantages of simulated annealing and tabu search. Furthermore, the greedy local search is used to find better neighborhood solutions for VRPTW. The Solomon’s problem instances are used for verifying the developed approach. Based on the number of vehicles required and the traveling distance, good results are obtained when the number of customers is equal to 25 and 50. In the problem with 100 customers, the developed approach finds all the best results in the C set. The results obtained form other problem sets are comparable with the existing state-of-art approaches. In many problems, 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 proposed approach can be used to solve the VRPTW at reasonable computation time.

**Keyword**: VRPTW; Hybrid Approach; Simulated Annealing; Tabu Search

## 1. 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 [44]. Many researches proposed exact methods and heuristics to solve this type of problem. Kolen et al. [33] developed a branch and bound approach to solve the VRPTW. Desrochers et al. [16] proposed a column generation approach that solved the Solomon's benchmark instances. Fisher et al. [17] 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 [10] were the first to propose the use of a savings algorithm to construct a feasible solution of vehicle routing problems. Miller and Gillett [21] 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 [48] 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 [43][11][1] 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 [48] 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 [9] proposed partial routes constructed and added a tabu list which is used to avoid cycling in the simulated annealing (SA). Osman [38] applied SA combined with tabu for VRPTW. Haibing and Andrew [24] proposed meta-heuristics based on SA and enhanced local search to solve VRPTW. Taillared et al. [49] performed tabu search based on GENIUS, cross exchange and or-opt for VRPTW. Potvin et al. [40] also used 2-opt and or-opt combined with tabu search for VRPTW. Gehring and Homberger [19][20] proposed hybrid approach to minimize the number of vehicle and total cost. Blanton and Wainwright [6] first proposed genetic algorithm with greedy algorithm for VRPTW. Bräysy et al. [7] ameliorated the search performance by improving genetic algorithm and evolution algorithm. Gambardella et al. [18] proposed a multiple ant colony system for vehicle routing problems with time windows ant system. These above algorithms provided versatile and effective solutions for VRPTW. Nevertheless, most solutions obtained are worse than the best solution found so far. In this paper, a hybrid approach is proposed to ameliorate the search performance for VRPTW. It takes the advantages of simulated annealing and tabu search. Additionally, the greedy local search is used to find better neighborhood solutions.

The remainder of this paper is organized as follows. Section 2 describes the VRPTW and the methods used to solve the VRPTW. Section 3 elaborates the proposed approach. In Section 4, computational results are compared to the solutions of the previous studies. Finally, conclusion and future research is included in the last section.

## 2. PROBLEM DEFINITION

Potvin and Bengio [39] 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 [51] 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 $i$,

$q_v$ loading 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

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

Subject to

$$\sum_{v=1}^{V}\sum_{j=1}^{N} x_{ijv} \le V \quad \text{for } i = 0 , \tag{2}$$

$$\sum_{j=1}^{N} x_{ijv} = \sum_{j=1}^{N} x_{jiv} \le 1 \quad \text{for } i = 0 \text{ and } v \in \{1, \dots, V\}, \tag{3}$$

$$\sum_{v=1}^{V}\sum_{j=0}^{N} x_{ijv} = 1 \quad \text{for } i \in \{1, \dots, N \tag{4}$$

$$\sum_{v=1}^{V}\sum_{i=0}^{N} x_{ijv} = 1 \quad \text{for } j \in \{1, \dots, N\}, \tag{5}$$

$$\sum_{i=0}^{N} m_i \sum_{j=0}^{N} x_{ijv} \le q_v \quad \text{for } v \in \{1, \dots, V\}, \tag{6}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} x_{ijv}(t_{ij} + f_i + w_i) \le r_v \quad \text{for } v \in \{1, \dots, V\}, \tag{7}$$

$$t_0 = w_0 = f_0 = 0, \tag{8}$$

$$\sum_{v=1}^{V}\sum_{i=0}^{N} x_{ijv}(t_i + t_{ij} + f_i + w_i) = t_j \quad \text{for } j \in \{1, \dots, N\}, \tag{9}$$

$$e_i \le (t_i + w_i) \le l_i \quad \text{for } i \in \{0, \dots, N\}, \tag{10}$$

Formula (1) is the objective function of the problem. The first set of constraints (2) specifies that there are at most $V$ routes going out of the depot. The second set of constraint (3) makes sure every route starts and ends at the delivery depot. The third set of constraints (4) and the forth set of constraint (5) restrict the assignment of each customer to exact one vehicle route. The fifth set of constraints (6) ensures the loading capacity of vehicle will not be violated. The sixth set of constraints (7) is the maximum travel time constraint. Other sets of constraints (8)–(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 research applied a hybrid approach which takes the advantages of simulated annealing and tabu search, and the greedy local search is used in finding neighborhood solution for solving 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. (3) TS can avoid cycling. 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, the hybrid SA-Tabu combined with the local search approach can more effectively find the (near) global optimum solution.

## 3. THE PROPOSED APPROACH

This study proposes a hybrid approach which takes the advantages of simulated annealing and tabu search for solving the vehicle routing problems with time windows. The ideas and characteristics of simulated annealing and tabu search are described as follows.

Introduced by Metropolis et al. [37] and popularized by Kirkpatrick et al. [30], the concept of simulated annealing is taken from nature. Annealing is the process through which slow cooling of metal produces good and low energy state crystallization, whereas fast cooling produces poor crystallization. The essential idea is to not to restrict the search algorithm to moves in solution-space that decrease the objective function (for an objective function we are trying to minimize), but to also (with some probability) moves that can increase the objective function. In principle, this allows a search algorithm to escape from a local minimum. 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.

Tabu search, initially suggested by Glover et al. [22][23], is an iterative improvement approach designed for getting (near) global optimum solutions to combinatorial optimization problems. The idea of TS can be described briefly as follows. Starting from an initial solution, TS iteratively moves from the current solution $X$ to its best improved solution $Y$ in the neighborhood of $X$, or if none exists, chooses the least worsening solution, until a superimposed stopping criterion becomes true. In order to avoid cycling to some extent, moves which would bring us back to a recently visited solution should be forbidden or declare tabu for a certain number of iterations. This is accomplished by keeping the attributes of the forbidden moves in a list, called a tabu list. The size of the tabu list must be large enough to prevent cycling, but small enough not to forbid too many moves. If a tabu move is better than the best solution obtained so far by the search, then this move can be selected even though it is tabu, overriding the tabu restriction. This is called the aspiration criterion**.**

For the application of the SA-Tabu approach to vehicle routing problem with time windows, the solution representation, the initial solution, objective function value calculation, the neighborhood, the tabu move, the aspiration criteria, the parameters used and procedures are discussed as follows.

### 3.1 Solution presentation and the initial solution

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 [12,17,15,10,9,16,5,0,11,2,20, 3,1,0,8,7,14,6,0,18,13,19,4] 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. The above four routes can be displayed graphically as shown in Figure 1. Each route starts from depot, visiting customers and ends at depot. This study used sequential insertion heuristic [48] 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.

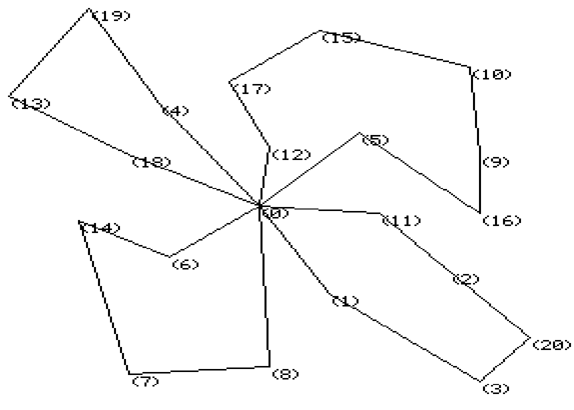


Figure 1. A graph representation of VRPTW solution.

### 3.2 Objective function value calculation

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 $C_1N + C_2$P($S$) + $C_3$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.

### 3.3 Neighborhood

The neighborhood solution of current solution can be generated either by 2-opt exchange or by insertion method. When SA-Tabu approach 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.

### 3.4 Tabu move and aspiration criterion

In order to avoiding cycling, the tabu_time matrix is applied to impose tabu moves. Let ($i$,$j$) element of the tabu_time matrix contain the iteration number at which job $i$ is allowed to return to the $j^{th}$ position of the scheduling sequence. The duration which a job is not allowed to move to a position of the scheduling sequence is called the tenure of tabu move. If the chosen neighborhood solution $Y$, of current solution $X$, belongs to the tabu moves, then solution $Y$ is discarded and the new neighborhood is generated until $Y$ does not belong to tabu moves, or $Y$ is the best solution found so far by the search.

### 3.5 Parameters used and the procedure:

The SA-Tabu begins with six parameters, namely $I_{iter}$, $T_0$, $T_F$, α, $MIN_T$ and $MAX_T$ where $I_{iter}$ denotes the number of iterations the search proceeds with a particular temperature, $T_0$ represents the initial temperature, $T_F$ represents the final temperature that stops the SA-Tabu procedure if the current temperature is lower than $T_F$, $\alpha$ is the coefficient controlling the cooling schedule, $MIN_T$ and $MAX_T$ are the minimal and maximal tenure of tabu moves, respectively. First, the current temperature $T$ is set to be the same as $T_0$. Next, an initial solution $X$ is randomly generated. The current best solution $X_{best}$ is set to be equal to $X$, the current objective function value $F_{cur}$ is set to be equal to the objective function value of $X$, $F_X$, and the best objective function value obtained so far $F_{best}$ is set to be equal to $F_{cur}$.

For each iteration, the next solution $Y$ is generated from $X$ either by swap or by insertion. The new solution $Y$ can not belong to tabu moves, unless new solution $Y$ is the best solution found so far. $T$ is decreased after running $I_{iter}$ iterations from the previous decrease, according to a formula $T \leftarrow \alpha T$, where $0 < \alpha < 1$. The tenure of tabu move is re-assigned by choosing an integral value between $MIN_T$ and $MAX_T$ randomly, when $T$ is decreased once.

Let obj($X$) denotes the calculation of the objective function value of $X$, and △ denote the difference between obj($X$) and obj($Y$); that is △=obj($Y$)-obj($X$). The probability of replacing $X$ with $Y$, where $X$ is the current solution and $Y$ is the next solution, given that △>0, is $e^{-\Delta/T}$. This is accomplished by generating a random number $r \in [0, 1]$ and replacing the solution $X$ with $Y$ if $r < e^{-\Delta/T}$. Meanwhile, if $\triangle \le 0$, the probability of replacing $X$ with $Y$ is 1. If the solution $X$ is replaced by $Y$, the tabu_time matrix is changed accordingly. If $T$ is lower than $T_F$, the algorithm is terminated. The $X_{best}$ records the best solution as the algorithm progresses. Following the termination of SA-Tabu procedure, the (near) global optimal schedule can thus be derived by $X_{best}$. The flowchart of the proposed SA-Tabu approach can be seen in the Figure 2.

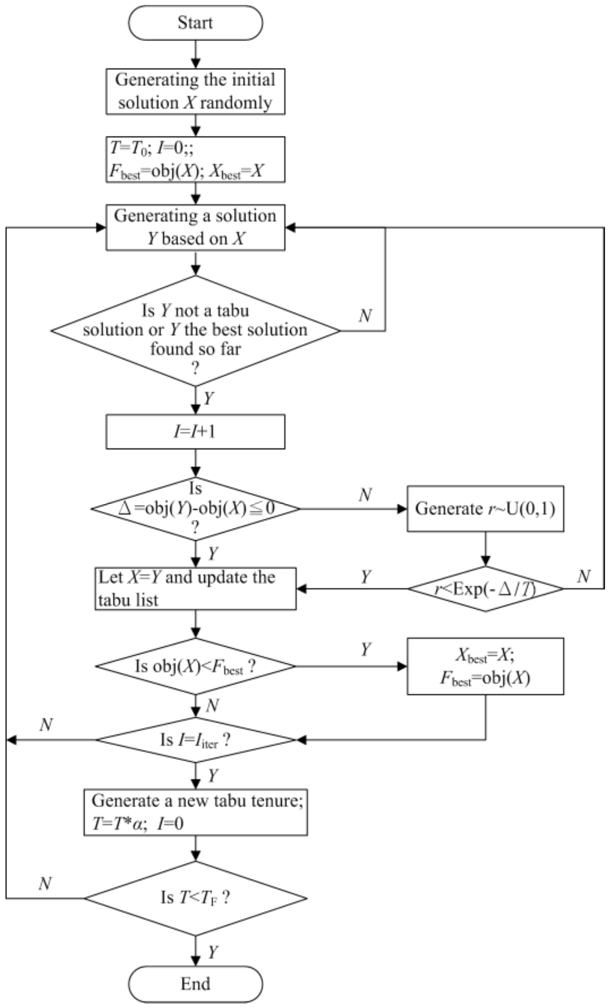


Figure 2. The flowchart of the proposed SA-Tabu approach.

## 4. Computational Results

Table 1. Our best solution and best published solution (with 25 customers).

| Problem | Best published (NV / TD/ Ref) | Our best solution (NV / TD) |
|---|---|---|
| C101 | 3/191.3/KDMSS | 3/191.81 |

| | | |
|---|---|---|
| C102 | 3/190.3/KDMSS | 3/190.74 |
| C103 | 3/190.3/KDMSS | 3/190.74 |
| C104 | 3/186.9/KDMSS | 3/187.45 |
| C105 | 3/191.3/KDMSS | 3/191.81 |
| C106 | 3/191.3/KDMSS | 3/191.81 |
| C107 | 3/191.3/KDMSS | 3/191.81 |
| C108 | 3/191.3/KDMSS | 3/191.81 |
| C109 | 3/191.3/KDMSS | 3/191.81 |
| C201 | 2/214.7/CR+L | 2/215.54 |
| C202 | 2/214.7/CR+L | 2/215.54 |
| C203 | 2/214.7/CR+L | 2/215.54 |
| C204 | 2/213.1/CR+KLM | 2/213.93 |
| C205 | 2/214.7/CR+L | **1/297.45** |
| C206 | 2/214.7/CR+L | **1/285.39** |
| C207 | 2/214.5/CR+L | 2/215.33 |
| C208 | 2/214.5/CR+L | **1/229.84** |
| R101 | 8/617.1/KDMSS | 8/618.33 |
| R102 | 7/547.1/KDMSS | 7/548.11 |
| R103 | 5/454.6/KDMSS | **4/473.39** |
| R104 | 4/416.9/KDMSS | 4/417.96 |
| R105 | 6/530.5/KDMSS | **5/556.72** |
| R106 | 5/465.4/KDMSS | **4/543.81** |
| R107 | 4/424.3/KDMSS | 4/425.27 |
| R108 | 4/397.3/KDMSS | 4/398.30 |
| R109 | 5/441.3/KDMSS | **4/460.52** |
| R110 | 4/444.1/KDMSS | 4/445.80 |
| R111 | 5/428.8/KDMSS | **4/429.70** |
| R112 | 4/393/KDMSS | 4/394.10 |
| R201 | 4/463.3/CR+KLM | **2/523.66** |
| R202 | 4/410.5/CR+KLM | **2/455.53** |
| R203 | 3/391.4/CR+KLM | **2/400.40** |
| R204 | 2/355.0/IV+C | 2/355.89 |
| R205 | 3/393/CR+KLM | **2/405.98** |
| R206 | 3/374.4/CR+KLM | **2/378.18** |
| R207 | 3/361.6/KLM | **2/362.79** |
| R208 | 1/328.2/IV+C | 1/329.33 |
| R209 | 2/370.7/KLM | 2/371.56 |
| R210 | 3/404.6/CR+KLM | **2/410.60** |
| R211 | 2/350.9/KLM | 2/351.91 |
| RC101 | 4/461.1/KDMSS | 4/462.16 |
| RC102 | 3/351.8/KDMSS | 3/352.74 |
| RC103 | 3/332.8/KDMSS | 3/333.92 |
| RC104 | 3/306.6/KDMSS | 3/307.14 |
| RC105 | 4/411.3/KDMSS | 4/412.38 |
| RC106 | 3/345.5/KDMSS | 3/346.51 |
| RC107 | 3/298.3/KDMSS | 3/298.95 |
| RC108 | 3/294.5/KDMSS | 3/294.99 |
| RC201 | 3/360.2/CR+L | **2/432.30** |
| RC202 | 3/338.0/CR+KLM | **2/376.12** |
| RC203 | 3/326.9/IV+C | **2/356.22** |
| RC204 | 3/299.7/C | **2/313.32** |
| RC205 | 3/338.0/L+KLM | **2/386.15** |
| RC206 | 3/324.0/KLM | **2/344.93** |
| RC207 | 3/298.3/KLM | **2/308.57** |
| RC208 | 2/269.1/C | **1/306.18** |

The developed approach of this study uses C programming language, computers equipped with Pentium IV 3.0G MHz CPU with 512 MB memory. In order to verify the effectiveness of the developed approach, the VRPTW benchmark problem instances provided by Solomon [48] 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. The number of customers is 25, 50 and 100 customers, respectively, for each problem set. 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 (C1 and C2) have clustered customers whose time windows were generated based on a known solution. Problems in set R (R1 and R2) have customers location generated uniformly randomly over a square. Problem in set RC (RC1 and RC2) have a combination of randomly placed and clustered customers.

Table 2. Our best solution and best published solution (with 50 customers).

| Problem | Best published (NV / TD/ Ref) | Our best solution (NV / TD) |
|---|---|---|
| C101 | 5/362.4/ KDMSS | 5/363.25 |
| C102 | 5/361.4/ KDMSS | 5/362.17 |
| C103 | 5/361.4/ KDMSS | 5/362.17 |
| C104 | 5/358/ KDMSS | 5/358.88 |
| C105 | 5/362.4/ KDMSS | 5/363.25 |
| C106 | 5/362.4/ KDMSS | 5/363.25 |
| C107 | 5/362.4/ KDMSS | 5/363.25 |
| C108 | 5/362.4/ CR+KLM | 5/363.25 |
| C109 | 5/362.4/ KDMSS | 5/363.25 |
| C201 | 3/360.2/ CR+L | **2/444.96** |
| C202 | 3/360.2/ CR+KLM | **2/403.81** |
| C203 | 3/359.8/ CR+KLM | **2/402.52** |
| C204 | 2/350.1/ KLM | 2/351.72 |
| C205 | 3/359.8/ CR+KLM | **2/430.03** |
| C206 | 3/359.8/ CR+KLM | **2/409.61** |
| C207 | 3/359.6/ CR+KLM | **2/398.34** |
| C208 | 2/350.5/ CR+KLM | 2/352.12 |
| R101 | 12/1044/KDMSS | **11/1100.72** |
| R102 | 11/909/KDMSS | **10/923.71** |
| R103 | 9/772.9/KDMSS | **8/784.76** |
| R104 | 6/625.4/KDMSS | 6/631.32 |
| R105 | 9/899.3/KDMSS | 9/914.31 |
| R106 | 8/793/KDMSS | 7/857.98 |
| R107 | 7/711.1/KDMSS | **6/735.27** |
| R108 | 6/617.7/CR+KLM | 6/620.26 |
| R109 | 8/786.8/KDMSS | **7/801.97** |
| R110 | 7/697/KDMSS | 7/699.38 |
| R111 | 7/707.2/CR+KLM | **6/756.66** |
| R112 | 6/630.2/CR+KLM | 6/637.52 |
| R201 | 6/791.9/CR+KLM | **2/967.96** |
| R202 | 5/698.5/CR+KLM | **2/813.35** |
| R203 | 5/605.3/IV+C | **2/666.64** |
| R204 | 2/506.4/IV | 2/509.25 |
| R205 | 4/690.1/IV+C | **2/740.73** |
| R206 | 4/632.4/IV+C | **2/658.34** |
| R207 | N/A | 2/588.38 |
| R208 | N/A | 2/490.05 |
| R209 | 4/600.6/IV+C | **2/659.07** |
| R210 | 4/645.6/IV+C | **2/668.72** |
| R211 | 3/535.5/IV+DP | **2/553.98** |
| RC101 | 8/944/KDMSS | 8/944.58 |
| RC102 | 7/822.5/KDMSS | 7/823.97 |
| RC103 | 6/710.9/KDMSS | 6/712.56 |
| RC104 | 5/545.8/KDMSS | 5/546.51 |
| RC105 | 8/855.3/KDMSS | 8/856.97 |

| | | |
|---|---|---|
| RC106 | 6/723.2/KDMSS | 6/724.65 |
| RC107 | 6/642.7/KDMSS | 6/643.86 |
| RC108 | 6/598.1/KDMSS | 6/599.17 |
| RC201 | 5/684.8/L+KLM | **3/838.55** |
| RC202 | 5/613.6/IV+C | **3/694.60** |
| RC203 | 4/555.3/IV+C | **2/674.40** |
| RC204 | 3/444.2/DP | **2/482.43** |
| RC205 | 5/630.2/IV+C | **3/761.38** |
| RC206 | 5/610/IV+C | **2/755.13** |
| RC207 | 4/558.6/C | **2/655.81** |
| RC208 | 5/684.8/L+KLM | **2/511.53** |

Table 3. Our best solution and best published solution (with 100 customers).

| 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 | 19/1651.26 |
| R102 | 17/1486.12/RT | 17/1490.66 |
| R103 | 13/1292.68/LL | 14/1218.42 |
| R104 | 9/1007.24/M | 10/985.30 |
| R105 | 14/1377.11/RT | 14/1383.67 |
| R106 | 12/1251.98/M | 12/1261.52 |
| R107 | 10/1104.66/S97 | 10/1125.707 |
| R108 | 9/960.88/BBB | 9/983.902 |
| R109 | 11/1194.73/HG | 12/1154.31 |
| R110 | 10/1118.59/M | 11/1097.03 |
| R111 | 10/1096.72/RP | 11/1057.09 |
| R112 | 9/982.14/GTA | 10/961.93 |
| R201 | 4/1252.37/HG | 4/1270.97 |
| R202 | 3/1191.70/RGP | 3/1258.53 |
| R203 | 3/939.54/M | 3/960.51 |
| R204 | 2/825.52/BH | 2/840.55 |
| R205 | 3/994.42/RGP | 3/1006.03 |
| R206 | 3/906.14/SSSD | 3/913.26 |
| R207 | 2/893.33/BVH | 2/924.11 |
| R208 | 2/726.75/M | 2/726.82 |
| R209 | 3/909.16/H | 3/926.46 |
| R210 | 3/939.34/M | 3/945.93 |
| R211 | 2/892.71/BH | 3/789.50 |
| RC101 | 14/1696.94/ TBGGP | 15/1640.32 |
| RC102 | 12/1554.75/ TBGGP | 13/1483.107 |
| RC103 | 11/1261.67/S98 | 11/1263.633 |
| RC104 | 10/1135.48/CLM | 10/1138.906 |
| RC105 | 13/1629.44/BBB | 14/1540.182 |
| RC106 | 11/1424.73/BBB | 12/1384.814 |
| RC107 | 11/1230.48/S97 | 11/1232.26 |
| RC108 | 10/1139.82/TBGGP | 10/1160.785 |
| RC201 | 4/1406.91/M | 4/1466.94 |
| RC202 | 3/1367.09/CC | 4/1161.29 |
| RC203 | 3/1049.62/CC | 3/1061.13 |
| RC204 | 3/798.41/M | 3/798.61 |
| RC205 | 4/1297.19/M | 4/1321.84 |
| RC206 | 3/1146.32/H | 3/1165.33 |
| RC207 | 3/1061.14/BH | 3/1093.76 |
| RC208 | 3/828.14/IKMUY | 3/844.47 |

Table 4. Comparisons of best averages on benchmark problems (with 100 customers).

| Reference | C1 | C2 | R1 | R2 | RC1 | RC2 |
|---|---|---|---|---|---|---|
| BBH [2] | 10.00 | 3.00 | 12.80 | 3.00 | 13.00 | 3.70 |
| | 828.47 | 590.60 | 1212.58 | 956.73 | 1379.86 | 1148.66 |
| BBB [5] | 10.00 | 3.00 | 12.17 | 2.73 | 11.75 | 3.25 |
| | 828.48 | 589.93 | 1230.22 | 1009.53 | 1397.63 | 1230.20 |
| CR [9] | 10.00 | 3.00 | 12.50 | 2.91 | 12.38 | 3.38 |
| | 909.80 | 684.10 | 1308.82 | 1166.42 | 1473.90 | 1401.50 |
| CLM [13] | 10.00 | 3.00 | 12.08 | 2.73 | 11.50 | 3.25 |
| | 828.48 | 589.93 | 1210.14 | 969.57 | 1389.78 | 1134.52 |
| GTA [18] | 10.00 | 3.00 | 12.38 | 3.00 | 11.92 | 3.38 |
| | 828.38 | 591.85 | 1210.83 | 960.31 | 1388.13 | 1149.28 |
| GH [20] | 10.00 | 3.00 | 12.00 | 2.73 | 11.50 | 3.25 |
| | 828.63 | 590.33 | 1217.57 | 961.20 | 1395.13 | 1139.37 |
| HA [24] | 10.00 | 3.00 | 12.08 | 2.91 | 11.75 | 3.25 |
| | 828.38 | 589.86 | 1215.14 | 953.43 | 1385.47 | 1142.48 |
| HG [25] | 10.00 | 3.00 | 11.92 | 2.73 | 11.63 | 3.25 |
| | 828.38 | 589.86 | 1228.06 | 969.95 | 1392.57 | 1144.43 |
| KCL [31] | 10.00 | 3.00 | 12.60 | 3.20 | 12.80 | 3.85 |
| | 833.32 | 593.00 | 1203.32 | 951.17 | 1382.06 | 1132.79 |
| LS [35] | 10.00 | 3.00 | 12.17 | 2.82 | 11.88 | 3.25 |
| | 841.33 | 591.03 | 1249.57 | 1016.58 | 1412.87 | 1204.87 |
| PB [39] | 10.00 | 3.00 | 12.60 | 3.00 | 12.10 | 3.40 |
| | 838.00 | 589.90 | 1296.8 | 1117.70 | 1446.20 | 1360.60 |
| RT [41] | 10.00 | 3.00 | 12.58 | 3.09 | 12.38 | 3.62 |
| | 828.45 | 590.32 | 1197.42 | 954.36 | 1369.48 | 1139.79 |
| RGP [42] | 10.00 | 3.00 | 12.08 | 3.00 | 11.63 | 3.38 |
| | 828.38 | 589.86 | 1210.21 | 941.08 | 1382.78 | 1105.22 |
| TBGGP [49] | 10.00 | 3.00 | 12.25 | 3.00 | 11.88 | 3.38 |
| | 828.45 | 590.30 | 1216.70 | 995.38 | 1367.51 | 1165.62 |
| THK [50] | 10.10 | 3.30 | 13.20 | 5.00 | 13.50 | 5.00 |
| | 861.00 | 619.00 | 1227.00 | 980.00 | 1427.00 | 1123.00 |
| TLZO [52] | 10.10 | 3.30 | 14.40 | 5.60 | 14.60 | 7.00 |
| | 860.62 | 623.47 | 1314.79 | 1093.37 | 1512.94 | 1282.47 |
| T [53] | 10.00 | 3.00 | 12.30 | 3.00 | 12.10 | 3.40 |
| | 830.89 | 640.86 | 1227.42 | 1005.00 | 1391.13 | 1173.38 |
| Z [54] | 10.00 | 3.00 | 12.80 | 3.00 | 13.00 | 3.70 |
| | 828.9 | 589.9 | 1242.70 | 1016.40 | 1412.00 | 1201.20 |
| Our best solution | **10.00** | **3.00** | 12.42 | 2.80 | 12.00 | 3.38 |
| | **828.38** | **589.86** | 1197.57 | 960.24 | **1355.50** | 1114.17 |

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 with 25, 50, 100 customers are about 20 to 25, 100 to110 and 230 to 240 seconds, respectively, using Pentium IV 3.0 GHz personal computer. In order to verify the performance of the developed approach, the results obtained are compared with other existing approaches. The results obtained for problems with 25, 50, and 100 customers are shown in Table 2, 3, and 4, respectively. In these tables, NV means number of vehicles used, TD represents travel distance, and Ref means the reference which obtained the corresponding best result.

The results for the six problem data sets with the number of customer equal to 25 are summarized in Table 1 for some of the best-reported heuristics for VRPTW, namely KDMSS [32], C [8], CR [12], KLM [29], IV [28], and L [34]. From simulation results, it shows that the proposed approach has better performance in C2, R1, R2, and RC1 problem sets. The results for the six problem data sets with the number of customer equal to 50 are summarized in Table 2 for some of the best-reported heuristics for VRPTW, namely KDMSS [32], C [8], CR [12], DP [15], KLM [29], IV [28] and L [34]。 From simulation results, it shows that the proposed approach has better performance in C2, R1, R2, and RC1 problem sets. Table 3 shows the best results obtained from the proposed approach and the best published results obtained so far by existing approaches for the number of customers equal to 100. The existing approaches include the follows: BH [3], BBB [5], CR [9], CC [14], CLM [13], GTA [18], HG [25], H [26], IKMUY [27], LL [24], LS [34], M [36], PB [39], RT [41], RGP [42], SSSD [45], S97 [46], S98 [47], TBGGP [49], THK [50], TLZO [52], T [53] and Z [54]. It can be noticed that the problems in set C all found known best solutions can be obtained by the developed approach, as shown in Table 3. Comparing to the results of previous studies, the developed approach found many 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. 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 4. 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.

## 5. CONCLUSIONS AND FUTURE RESEARCH

This research used the sequential insertion heuristic to obtain the initial feasible solution of VRPTW and then utilized hybrid SA-Tabu approach 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 other sets 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 depots 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] Baker, E. 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.

[2] Beatrice, O., 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.

[3] Bent, R., 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.

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

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

[6] Blanton, J. 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.

[7] Bräysy, O., 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.

[8] Chabrier, A., "Vehicle routing problem with elementary shortest path based column generation." Forthcoming in: Computers and Operations Research, 2005.

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

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

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

[12] Cook, W. and J. L. Rich, "A Parallel Cutting Plane Algorithm for the Vehicle Routing Problem with Time Windows," Technical Report, TR99-04, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

[13] Cordeau, J. F., 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.

[14] Czech, Z. J. 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.

[15] Danna, E. and C. Le Pape, "Accelerating branch-and-price with local search: a case study on the vehicle routing problem with time windows," In: Column Generation, G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), Kluwer Academic Publishers, 2005, pp.99-130.

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

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

[18] Gambardella, L., 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.

[19] Gehring, H. 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.

[20] Gehring, H. 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.

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

[22] Glover, F., "Tabu search-Part I," ORSA Journal on Computing, vol. 1 No. 3, 1989, pp. 190-206.

[23] Glover, F., "Tabu search-Part II," ORSA Journal on Computing vol. 2, No.1, 1990, pp. 4-32.

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

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

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

[27] Ibaraki, T., 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.

[28] Irnich, S. and D. Villeneuve, "The shortest path problem with k-cycle elimination (k≦3): improving a branch-and-price algorithm for the VRPTW." Forthcoming in: INFORMS Journal of Computing, 2005.

[29] Kallehauge, B., J. Larsen, and O. B. G. Madsen, "Lagrangean duality and non-differentiable optimization applied on routing with time windows - experimental results." Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.

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

[31] Kit, H.W., Chin, J. and Lim, A., "A Hybrid Genetic Algorithm for the Vehicle Routing Problem," International Journal on Artificial Intelligence Tools (to appear).

[32] Kohl, N., J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis, "2-Path cuts for the vehicle routing problem with time windows," Transportation Science, vol. 33, 1999, pp. 101-116.

[33] Kolen, A. W. J., A. H. G.R. Kan, and H. W. J. M. Trienekens, “Vehicle routing with time windows,” Operations Research, vol. 35, 1987, pp.266-273.
[34] Larsen, J., “Parallelization of the vehicle routing problem with time windows ” Ph.D. Thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.
[35] Liu, F.-H. 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.
[36] Mester, D., “An evolutionary strategies algorithm for large scale vehicle routing problem with capacitate and time windows restrictions,” In Conf. Mathematical and Population Genetics, University of Haifa, Israel, 2002.
[37] Metropolis, N., 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.
[38] Osman, I. H., “Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems,” Annals of Operations Research, vol. 42, 1993b, pp. 421-451.
[39] Potvin, J. Y. and S. Bengio, “The vehicle routing problem with time windows Part Ⅱ: Genetic Search,” Informs Journal on Computing, vol. 8, 1996, pp.165-172.
[40] Potvin, J. Y., 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.
[41] Rochat, Y. and E. D. Taillard, “Probabilistic diversification and intensification in local search for vehicle routing,” Journal of Heuristics, vol. 1, 1995, pp. 147-167.
[42] Rousseau, L.-M., 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.
[43] Russell, R. A., “An effective heuristic for the m-tour traveling salesman problem with some side conditions,” Operations Research, vol. 25, 1977, pp. 517-524.
[44] Savelsbergh, M., “Local search for routing problems with time windows,” Annals of Operations Research, vol. 4, 1985, pp. 285-305.
[45] Schrimpf, G., 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.
[46] Shaw, P., “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.
[47] Shaw, P., “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.
[48] Solomon, M. M., “Algorithms for the vehicle routing and scheduling problems with time windows constraints,” Operations Research, vol. 35, 1987, pp. 254-265.
[49] Taillard, E., 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.
[50] Tan, K. C., 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.
[51] Tan, K. C., 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.
[52] Tan, K.C., Lee, L. H., Zhu, Q. L. and Ou, K., “Heuristic Methods for Vehicle Routing Problem with Time Windows,” Artificial Intelligent in Engineering, vol. 15, 2001, pp. 281–295.
[53] Thangiah, S.R., “A Hybrid Genetic Algorithms, Simulated Annealing and Tabu Search Heuristic for Vehicle Routing Problems with Time Windows,” in Practical Handbook of Genetic Algorithms, Volume III: Complex Structures, edited by L. Chambers, CRC Press, 1999, pp. 347–381.
[54] Zhu, K. Q., “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.