A multi-criteria vehicle routing problem with soft time windows by simulated annealing

R. Tavakkoli-Moghaddam*

Department of Industrial Engineering, Faculty of Engineering, University of Tehran, Iran

N. Safaei

Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

M. A. Shariat

Department of Industrial Engineering, Graduate School, Islamic Azad University, Tehran South Branch, Tehran, Iran

Abstract

This paper presents a multi-criteria vehicle routing problem with soft time windows (VRPSTW) to minimize fleet cost, routes cost, and violation of soft time windows penalty. In this case, the fleet is heterogeneous. The VRPSTW consists of a number of constraints in which vehicles are allowed to serve customers out of the desirable time window by a penalty. It is assumed that this relaxation affects customer satisfaction and penalty is equal to a degree of customer dissatisfaction. The VRP, which is an extension of traveling salesman problem (TSP), belongs to a class of NP-hard problems. Thus, it is necessary to use meta-heuristics for solving VRP in large-scale problems. This paper uses a simulated annealing (SA) approach with 1-Opt and 2-Opt operators for solving the proposed mathematical model. The proposed model is then solved by the Lingo software and the associated solutions are compared with the computational results obtained by the SA approach for a number of instance problems. The obtained results are promising and indicating the efficiency of the proposed SA approach.

Keywords: Multi-criteria vehicle routing problem; Mathematical model; Time windows; Simulated annealing

1. Introduction

In the classical vehicle routing problem (VRP), routes are constructed to dispatch a fleet of homogenous or heterogeneous vehicles to serve a set of customers from a single distribution depot. Each vehicle has a fixed capacity and each customer has a known demand that must be fully satisfied. Each customer must be serviced by exactly one visit of a single vehicle. The total demands of the customers serviced by the vehicle must not exceed the capacity of the vehicle. Each vehicle must depart from and return to the depot. A vehicle routing problem with time windows (VRPTW) is an extension of the classical VRP in which each customer serviced in a specified time interval named time window. If a vehicle arrives at a station (or node) to pick up a customer earlier than the lower bound of the customer's time window, the vehicle must wait until the service is possible. Also, if a vehicle arrives later than the upper bound of the customer's time window, the vehicle cannot serve the customer. The VRPTW has a great number of practical applications in industries and services such as the distribution of cash amounts among bank branches, disposal of garbage and industrial wastes, distribution of fuel to and among fuel stations, school transportation services and the like. The VRPTW can be classified into two general categories known as hard and

---

* Corresponding author. E-mail: tavakoli@ut.ac.ir

soft VRPTW. These two categories are explained below:

- **Hard VRPTW:** In hard VRPTW, vehicles are expected to fulfill a customer service within a specified time window as shown in figure 1. In other words, each customer i has an allowed service time interval $[a_i, b_i]$ as shown in figure 2, in which the service must occur. The service out of this interval is not allowed. figure 3 depicts customer satisfaction in a hard time window.

- **Soft VRPTW:** In soft VRPTW, each customer i has a desirable time window with the high satisfaction as $[a_i, b_i]$ and a hard time window as $[LB_i, UB_i]$ where $LB_i < a_i$ and $UB_i > b_i$. An instance of soft time window is shown in figure 4. In intervals $[LB_i, a_i)$ and $(b_i, UB_i]$, the service is allowed but a penalty is set for the service. As shown in figure 5, the customer satisfaction in interval $[a_i, b_i]$ is maximum and in intervals $[LB_i, a_i)$ and $(b_i, UB_i]$ is reducing.

The VRPTW belongs to the class of the NP-hard combinatorial optimization problems [13]. Although optimal solutions can be obtained by using exact and numerous methods, the computation time required to solve the VRPTW is optimally prohibitive [7]. Since heuristic methods often produce near-optimal solutions in a reasonable amount of computational time, most of the researches have focused on the design of heuristics and metaheuristics [1, 2, 5, 6, 20]. The VRPTW in particular is still ''much more difficult'' to solve than the classical VRP [17]. Hence, primarily heuristic procedures are suggested for larger-sized instances of the VRPTW. In the last decade, quite good results have been achieved for the VRPTW by metaheuristics. Two groups of metaheuristics seem to be appropriate for solving the VRPTW: (1) Metaheuristics controlling local search processes, such as tabu search [4, 18, 22], simulated annealing [3], genetic algorithms [16, 24], evolution strategies [10], large neighborhood search [19], and guided local search [11]; and (2) metaheuristics controlling a subordinate construction heuristic, such as the greedy randomized search procedure [14], the RNET meta-heuristic [12], and multiple ant colony systems [14].

Only a few authors report on the solution to the VRPTW by means of hybrid approaches. Gambardella, *et. al.* [8] and Glover, *et. al.* [9] have pointed out that hybrid approaches focus on enhancing the strengths and compensating for the weaknesses of two or more complementary approaches.

The aim is to generate better solutions by combining the key elements of competing methodologies. Osman and Kelly [15] have proposed a two-phase hybrid metaheuristic method for the VRPTW based on this idea. The best individual is determined by a genetic algorithm in which the first search phase is passed over to a tabu search algorithm in order to improve the achieved solution in the second search phase. On the other hand, the tabu search algorithm includes an element of the simulated annealing concept. This element is used to control the selection of neighborhood solutions. Thangiah, *et. al.* [25] have proposed a hybrid method which repeatedly carries out two subsequent steps. Partial routes constructed in the first step are improved by the simulated annealing in the second step. In the simulated annealing steps, a tabu list is used to avoid cycling.

In this paper, the authors propose a new mathematical model of soft VRPTW minimizing fleet cost, total traveled distances cost, and penalty of violating soft time windows. It is assumed that the fleet is heterogeneous. This type of soft VRPTW is used in most real-world situations such as school services and staff transportation scheduling. The proposed model is solved by the simulated annealing algorithm in large-scale problems.

This paper is organized as follows. The problem formulation is described in Section 2. The structure of the SA algorithm is explained in Section 3 with respect to the proposed model. The computational results are shown in Section 4 and the conclusion is presented in Section 5.
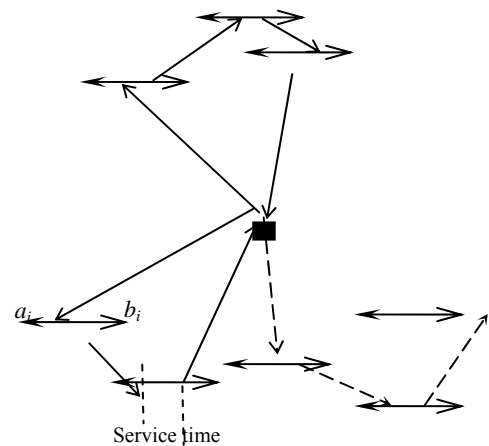

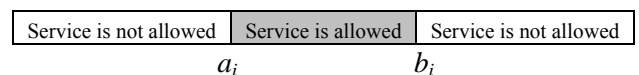**Figure 1**. Vehicle routes with hard time window.

Service is not allowed	Service is allowed	Service is not allowed
	a_i	b_i

Figure 2. Service time interval $[a_i, b_i]$ in hard time window.
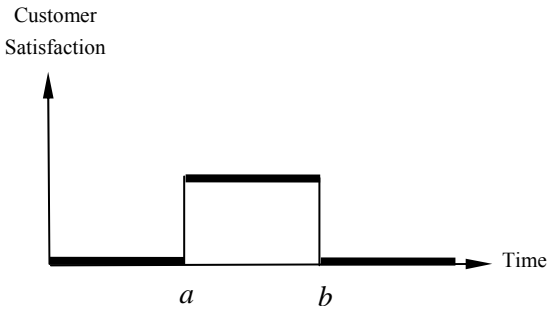
Figure 3. Customer satisfaction in hard time window.

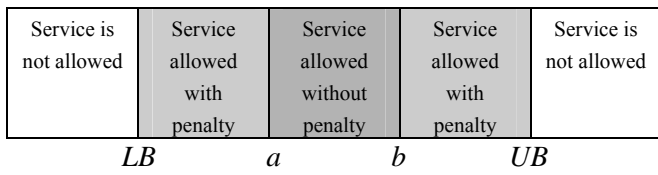| Service is not allowed	Service allowed with penalty	Service allowed without penalty	Service allowed with penalty	Service is not allowed
LB	a		b	UB

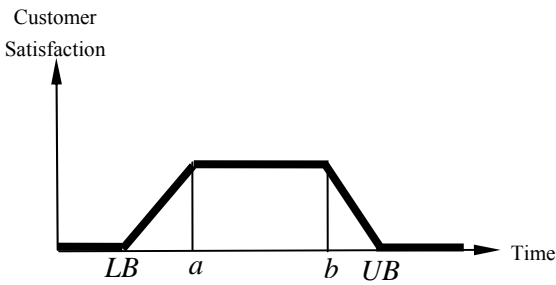**Figure 4**. Soft time window.

Figure 5. Customer satisfaction in soft time window.

2. Problem formulation

The problem is defined as follows: Let $G(V,E)$ be a complete graph, where $V=\{1,2,..,i,…,N\}$ is the node set and $E=\{(i,j): i,j \in V, i \neq j\}$ is the arc set. Node $i=1$ represents a depot while the remaining nodes are corresponded to the demand points or customers. Each node i has a desirable service time interval $[a_i, b_i]$ and a hard time window as $[LB_i, UB_i]$. The matrix $D = \{L_{ij} : (i,j) \in E, i,j \in V, L_{ij}>0\}$ denotes the physical distance between nodes. Also, let $F = \{1,2,…, M\}$ be fleet or set of available vehicles. Each vehicle j has capacity C_v and cost B_v where B_v / C_v is fixed.

The problem is solved under the following assumptions:

1. Each node is visited only once by a single vehicle.
2. Each vehicle must start and end its route at the depot.
3. Total demand served by each vehicle cannot exceed its capacity.
4. Each node must be served in related hard time window.
5. The mean velocity of travel for all vehicles is constant.
6. A service (wait) time in each node is allowed.
7. The hard time windows must not exceed.
8. The soft time windows can be violated at a fixed cost.
9. The fleet is heterogeneous and there are three types of capacity as: small, medium, and large for vehicles. Also, the cost of each unit of capacity is fixed for all vehicle types.

2.1. Definition parameters

A number of parameters are used in the proposed mathematical model as follows:

N: Number of demand nodes or stations (depot is at the node $i=1$).
M: Number of vehicles available.
C_v: The capacity of the vehicle v.
q_i: Demand at the node i. ($q_1= 0$)
B_v: Cost of the vehicle v, $B_v = \delta \times C_v$ where $\delta =$ cost of each unit of capacity.
a: Mean traveling speed of each vehicle.
d_{ij}: Distance between nodes i and j where $d_{ii}= M$. ($M=$ an arbitrary big number).
t_{ij}^v: Time for rendering service to the node plus the time. For covering the distance from node i to j by the vehicle v. This parameter has appeared in the model as a function of distance and speed.
g: Cost of a unit travel by the each vehicle.
s_i^v: Start time of service in node i by the vehicle v.
P_e: Unit penalty of earliness.
P_l: Unit penalty of lateness.
a_i: Lower bound of soft time window for the node i
b_i: Upper bound of soft time window for the node i
y_{vi}^e: $\max\{e_i-s_i^v, 0\}$ is the violation degree of lower bound of soft time.
y_{vi}^l: $\max\{s_i^v-l_i, 0\}$ is the violation degree of upper bound of soft time.
LB_i: Lower bound of hard time window for the node i
UB_i: Upper bound of hard time window for the node i

2.2. Definition of decision variables

Two decision variables are defined as z_v, x^v_{ij} as follows:

$$x^v_{ij} = \begin{cases} 1 & \text{if arc } (i,\ j) \in E \text{ is traversed by vehicle } v \\ 0 & \text{otherwise} \end{cases}$$

$$z_v = \begin{cases} 1 & \text{if vehicle } v \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

2.3. Mathematical formulation

By considering the defined parameters and variables, the proposed mathematical model consists of a multi-criteria function and 15 constraints for which we will give more detailed explanations after presenting the model.

$$\text{Min} \left[\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{v=1}^{M} g \times d_{ij} \times x^v_{ij} \right] + \left[\sum_{v=1}^{M} B_v \times z_v \right] + \left[\sum_{i=1}^{N} p_e y^e_{ki} + p_l y^l_{ki} \right] \quad (1)$$

Subject to:

$$\sum_{i=2}^{N} x^v_{i1} = 1 \quad \forall v \quad (2)$$

$$\sum_{j=2}^{N} x^v_{1j} = 1 \quad \forall v \quad (3)$$

$$\sum_{i=1}^{N}\sum_{v=1}^{M} x^v_{ij} = 1 \quad \forall j \quad (4)$$

$$\sum_{j=1}^{N}\sum_{v=1}^{M} x^v_{ij} = 1 \quad \forall i \quad (5)$$

$$\sum_{i=1}^{N} x^v_{ik} - \sum_{j=1}^{N} x^v_{kj} = 0 \quad \forall v, k \quad (6)$$

$$\sum_{i=1}^{N} q_i \left(\sum_{j=1}^{N} x^v_{ij} \right) \leq C_v \quad \forall v \quad (7)$$

$$M(1 - x^v_{ij})s^v_j \geq s^v_i + \frac{L_{ij}}{a} + \lambda d_i \quad \forall i,j,v \quad (8)$$

$$S^v_i \geq LB_i \quad (9)$$

$$S^v_i \leq UB_i \quad (10)$$

$$z_v = 1 - x^v_{11} \quad \forall v \quad (11)$$

$$z_v \geq x^v_{ij} \quad \forall v,i,j \quad (12)$$

$$y^e_{ki} \geq b_i - s^v_i \quad (13)$$

$$y^l_{ki} \geq s^v_i - a_i \quad (14)$$

$$\sum_{v=1}^{V}\sum_{i=1}^{N}\sum_{j=1}^{N} x^v_{ij} \geq r(S) \quad \forall S \subseteq V - \{0\}, S \neq \varnothing \quad (15)$$

$$x^v_{ij}, z_v \in \{0,1\} \quad (16)$$

The objective function (1) of the proposed model consists of three components. The first component is equal to the sum of traveled distances by all the used vehicles. The second component is defined as the fleet cost that is equal to the sum of costs related to the used vehicles. The third component of objective function is equal to total penalty of outrage from soft time windows in all demand points.

The constraints (2) and (3) impose that start and end of route for each vehicle must be at the depot. The constraints (4), (5), and (6) indicate that exactly one vehicle enters and leaves each node and serves it. The constraint (7) assures that the vehicles' capacity is not exceeded. The constraint (8) determines the start time of service for each vehicle in covered nodes. The constraints (9) and (10) assure that the start time of services does not violate the hard time windows. The constraints (11) and (12) determine the vehicles that are used. If vehicle v is not used then $x^v_{11}=1$, else $x^v_{11}=0$. The constraints (13) and (14) determine the violation degree of the lower and upper bound of soft time windows for each node. The constraint (15) that named capacity-cut constraint ensures that the sub-tours are eliminated. The constraint (16) guarantees the decision variables are zero or one.

2.4. Service time consideration

The constraint (8) forces the model to consider a service time (ST) in each node according to the equation (17). The service time can implicate to wait time for persons or a change of speed between current and succeeding node. It can influence a reduction of soft windows violation in the third term of the objective function.

$$S^v_j \geq S^v_i + \left(\frac{d^v_{ij}}{a} \right) - M(1 - x^v_{ij})$$

$$\Rightarrow ST_j = S^v_j - \left[S^v_i + \left(\frac{d^v_{ij}}{a} \right) - M(1 - x^v_{ij}) \right] \quad (17)$$

3. Simulated annealing

Simulated annealing (SA) is a stochastic relaxation technique that has its origin in statistical mechanics

[21, 23]. It is an approach based on the Mont Carlo's model that is used for the study of the relation among atomic structure, entropy, and temperature through the cooling process of a metal or material. The physical process of cooling that used for reducing temperature of a material to a minimal degree of energy is called thermal equilibrium. Cooling process is commenced with a material in a state of fusion and the temperature is gradually decreased. A metal or material may approach thermal equilibrium at any temperature. The temperature must not be subject to a rapid decrease; otherwise the substance will not approach a minimal energy state. Temperature reduction is similar to the objective value reduction in a minimization problem that is carried out by a series of improving variations. In order to allow a slow decrease of temperature, non–improving variations in the objective function must be made with a certain probability in a way that such a probability is also reduced when a reduction is made in the objective value. This prevents the algorithm from being entrapped by local optimums. Thus, temperature will act as a control parameter in an optimization problem.

SA uses a stochastic approach to direct the search. It allows the search to proceed to a neighboring state even if the move causes the value of the objective function to become worse. SA guides the original local search method in the following way. If a move to a neighbor X' in the neighborhood $N(X)$ decreases the objective function value or leaves it unchanged, then the move is always accepted. More precisely, the solution X' is accepted as the new current solution if $\Delta \leq 0$, where $\Delta = C(X') - C(X)$ and $C(X)$ is the objective function value. Moves, which increase the objective function value, are accepted with a probability of $e^{(-\Delta/T)}$ to allow the search to escape a local optimum; where T is a parameter named the temperature. The value of T varies from a relatively large value to a small value close to zero. These values are controlled by a cooling schedule that specifies the initial and incremental temperature values at each stage of the algorithm.

3.1. Initial solution generation

The first step in each metaheuristic approach is to generate initial solutions. The following innovative approach has been used for generating initial solutions.

Let S be a set of no served nodes; $S \subset V - \{1\}$ and V be a set of nodes. The depot is represented by the node 1.
Let F_0 be a set of no used vehicles; $F_0 \subset F$ and F be a set of fleet.
Let C_v be the capacity of vehicle v.
Let $C_{max} = max_{v \in F_0} \{C_v\}$ (maximum capacity in fleet).

Steps of complete algorithm are follows:
1. Let $S = V - \{1\}$ and $q = 0$, where q is a counter of used capacity of the current vehicle.
2. Select node i at random where $i \in S$ and let $q = d_i$.
3. Allocate vehicle $v \in F_0$ to node i where $C_v = C_{max}$.

    a) Find a node j at random where $j \in S$ and generate a service time ST_j randomly according to Section 2.4. Then dispatch vehicle v to node j to serve in which constraints (7), (9), and (10) have not violated.
    b) **IF** node j is not found **THEN** dispatch vehicle v to depot **AND** let $v = v^*$ where $|C_{v^*} - q| = min_{v \in F_0} \{|C_v - q|\}$ **AND** set $F_0 = F_0 - \{v^*\}$ **ELSE** let $S = S - \{j\}$ **AND** $q = q + d_j$.
4. Repeat step 4 until vehicle v return to the depot.
5. Repeat steps 2 to 5 until $S = \varnothing$.

3.2. Generation of neighborhood solution

Two efficient operators have been used to search the neighboring solutions within the feasible space. These operators are explained bellow:

2-Opt operator. In this type of operator, two routes belonging to two vehicles are randomly chosen from the existing feasible answer and then two nodes out of the two routes are exchanged with each other with the observance of vehicle capacity and service time constraints.

1-Opt operator. In this type of operator, two routes of two vehicles are randomly chosen out of the existing feasible answer and then a node is deleted from a route and is added to the other route with the observance of vehicle capacity and service time constraints. Figures 6 and 7 depict a profile of the two presented operators used in solving problems.
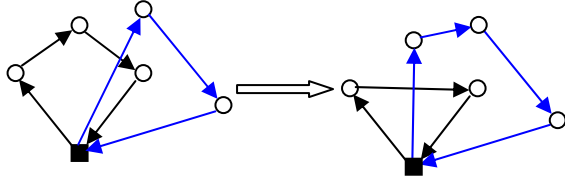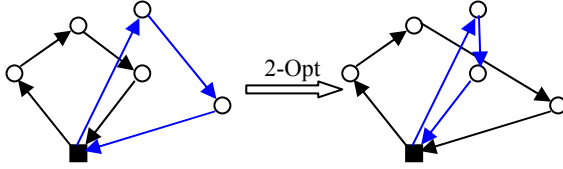
Figure 6. Implementation of operator 1-opt.

Figure 7. Implementation of operator 2-opt.

3.3. SA algorithm

The SA has two inside and outside loops. The inside loop controls the achievement to equipment in the current temperature and outside loop controls the decreasing rate of temperature. The SA parameters are listed bellow:

EL (Epoch length): Number of accepted solutions achieved in each temperature.

MTT: Maximum number of consecutive temperature trails.

T_0.: Initial temperature.

α : Rate of decreasing the current temperature (cooling schedule).

X : A feasible solution.

$C(X)$: The objective function value in respect to X.

n : Counter of number of accepted solutions in each temperature.

r : Counter of number of consecutive temperature trails, where T_r is equal to temperature in iteration r. The steps of the SA algorithm are shown in figure 8.

4. Computational results

Computational results have been shown in two cases which have been devoted to the model verification and the SA results respectively.

For the model verification, fourteen test problems are solved by the Lingo 6.0 software. To decrease the model complexity, some of parameters are considered constant as a priori shown in table 1. It is also assumed that the fleet has three types of capacities as small (20 persons), medium (30 persons) and large (40 persons) sizes. The values of hard time windows boundaries (LB_i and UB_i) are considered constant. The

value of soft time windows boundaries (a_i and b_i) are randomly generated in intervals $[LB_i, UB_i]$. The value of demand in nodes is randomly generated at random in uniform distribution $[0,20]$ and $[0,10]$. The distances between nodes are generated at random in uniform distribution $[0,100]$. The CPU times are correspond to an Intel® Celeron® mobile 1.3 GHz processor.

The comparison of optimal and SA solutions in small-sized problems is shown in table 2. As depicted in table 2, the proposed model is sensitive to *UB* value. By decreasing of *UB* parameter, the CPU time increases progressively for such problems (7,8), (9,10) and (11,12). The column 'Var' in table 2 indicates the number of variables in the model and the column 'Cons' indicates the number of constraints. Also, the column 'O.F.V' presents the objective function value. The column 'k' depicts the number of used vehicles in each problem. For instance in problem 1, the number of available vehicles is equal to 6 and the number of used vehicles is equal to 3. The column 'B&B' indicates the number of branches in a branch-and-bound method. The complexity of the proposed model with respect to number of variables, constraints, CPU time, and branches is shown in figure 9. The average gap between optimal and SA solutions is 4.8 percent implicating the efficiency of the SA algorithm. Also, the CPU times is related to SA solutions have egregious difference in respect to optimal ones.

$r = 0$ $,X^{best} = \varnothing$

Generate X^0

$X^{best} = X^0$

Do (*Out Side loop*)

  $n = 0$

  Do (*In Side loop*)

    Select an operator (1-Opt or 2-Opt move) randomly and run over

    X^n operator

as : $X^n \rightarrow X^{new}$

    $\Delta C = C(X^{new}) - C(X^{best})$

    If $\Delta C < 0$ **Then**

      $X^{best} = X^{new}$ and $n = n +1$ and $X^n = X^{new}$

    Else

      Generate $y \rightarrow U(0,1)$ Randomly

      Set $Z = e^{-\Delta C/T_r}$

      If $y < z$ **Then** $n = n +1$ and $X^n = X^{new}$

    End if

  Loop While($n < EL$)

  $r = r + 1$

  $T_r = T_{r-1} - \alpha \times T_{r-1}$

Loop While ($r < MTT$ and $T_r > 0$)

Print X^{best}

Figure 8. SA algorithm.

For verifying the SA algorithm in large-scale problems, the results obtained from SA are compared with the lower bound solution (LBS) by five test problems, as shown in table 3. The LBS is calculated by the relaxation of constraints (9) and (10) in the proposed mathematical model. In other words, the LBS is obtained by solving a classical VRP without time windows. The efficiency of SA can be measured by comparing initial and final solutions reported in table 3. The final solution is yielded about one hour. The detailed components of the objective function for problem 5 in table 3 is shown in figure 9. As shown in figure 10, the fleet and route costs are approximately constant during annealing process rather than the penalty of violating soft time windows.

Figure 9. Complexity of the proposed model.
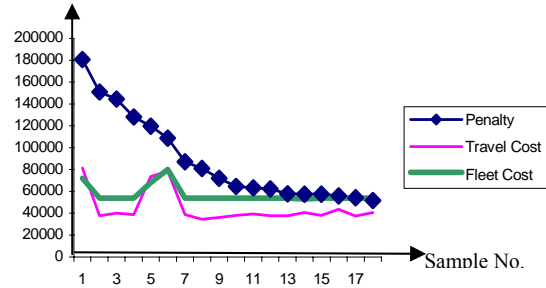
Figure 10. Cost details for the problem 5 from Table 3.

Table 1. Parameter settings.

P_e	P_l	g	δ	EL	MTT	T_0	T_f	α
10	20	2	100	100	100	5	0	0.99

Table 2. Comparison of optimal and SA solutions in small-sized problems.

Problem			Optimal					SA		Gap
No.	$N\times M\times UB$	Var	Cons	O.F.V	CPU Time	K	B&B	O.F.V	CPU Time (Sec.)	(%)
1	6×3×50	1572	275	4887	00:00:05	3	509	4887	0.8	0
2	6×3×60	1572	275	4176	00:00:04	3	537	4176	3.7	0
3	6×3×70	1572	275	4000	00:00:08	3	708	4000	11.5	0
4	6×3×80	1572	275	4090	00:00:08	3	1087	4090	5.6	0
5	6×3×90	1572	275	3120	00:00:05	2	595	3120	0.9	0
6	6×3×100	1572	275	3319	00:00:07	2	733	3344	17.2	0.7
7	7×3×80	2421	361	4105	00:00:38	3	1504	4105	16.6	0
8	7×3×100	2421	361	3889	00:00:31	2	1471	4105	1.16	5.5
9	8×3×80	3534	459	6817	00:04:04	3	8938	7075	0.2	3.7
10	8×3×100	3534	459	4225	00:01:03	3	2287	4225	0.1	0
11	9×2×120	3289	386	7449	00:12:52	2	20036	7529	0.1	1
12	9×3×120	4947	569	4230	00:59:28	3	81758	4601	110	8.7
13	10×2×150	4465	468	3795	01:15:00	2	84779	4712	0.5	24
14	10×3×150	6691	691	3795	02:00:00	3	135728	4712	0.7	24

Table 3. Comparison of lower bound and SA solutions in large-sized problems.

Test Problem		Lower Bound		SA Solution					
				Initial Solution		Final Solution			
No.	$N\times M\times UB$	O.F.V	Used Vehicles	O.F.V	CPU Time (Sec.)	O.F.V	CPU Time (Sec.)	Used Vehicles	
1	20×7×140	4025	4	29861	0	12117	124	6	
2	20×7×180	4025	4	41773	0	11382	109	4	
3	50×17×200	9625	8	107047	0	37276	347	9	
4	50×17×500	9625	8	298533	0	80029	604	8	
5	100×30×500	18550	18	767785	0	234968	955	14	

5. Conclusions

In this paper, the authors propose a multi-criteria model of a vehicle routing problem with time windows (VRPSTW) with heterogeneous fleet of fixed cost and service time in nodes. The objective is to minimize the fleet cost, routes cost, and degree of violating soft time windows boundaries. The proposed model is solved by the Lingo 6.0 and SA. The associated computational results have been reported and compared. The efficient SA algorithm based on virtual service time is used for generating initial solutions. Two operators 1-Opt and 2-Opt are used to improve the quality of obtained solutions. These solutions show that the proposed model is verifiable and confinable and the proposed SA is a suitable approach to solve such a hard model.

References

[1] Braysy, O. and Gendreau, M., 2002, Vehicle routing problem with time windows, part I: route construction and local search algorithms. *Transportation Science*, In press.

[2] Braysy, O. and Gendreau, M., 2002, Vehicle routing problem with time windows - part II: metaheuristics. *Transportation Science,* In press.

[3] Chiang, W. C. and Russell, R. A., 1996, Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63, 3–27.

[4] Chiang, W. C. and Russell, R. A., 1997, A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal on Computing* 9, 417-430.

[5] Cordeau, J.-F., Desaulniersm, G., Desrosiers, J., Solomon, M. M. and Soumis, F., 2001, The VRP with time windows. In: Toth P, Vigo D, editors. The vehicle routing problem, *SIAM monographs on discrete mathematics and applications*, Philadelphia: SIAM, 157–94.

[6] Desrosiers, J., Dumas, Y., Solomon, M. M. and Soumis, F., 1995, Time constrained routing and scheduling. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL, editors. *Handbooks in operations research and management science* 8: Network Routing, Amsterdam: Elsevier, 1995, 35–139.

[7] Descrochers, M., Desrosiers, J. and Solomon, M. M., 1992, A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40, 342-354.

[8] Gambardella, L. M., Taillard, E. and Agazzi, G., 1999, *MACSVRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows*. Technical Report IDSIA-06-99, Lugano, Switzerland.

[9] Glover, F., Kelley, J. P. and Laguna, M., 1995, Genetic algorithms and tabu search: Hybrids for optimization. *Computers and Operations Research*, 22, 111-134.

[10] Homberger, J. and Gehring, H., 1999, Two evolutionary metaheuristics for the vehicle routing problem with time windows. In: Laporte, G., Semet, F. (Eds.), Metaheuristics for location and routing problems, *Information Systems and Operational Research*, 37, 297-318.

[11] Kilby, P., Prosser, P. and Shaw, P., 1999, Guided local search for the vehicle routing problem with time windows. In: Voss, S., Martello, S., Osman, I.H., Roucairol, C. (Eds.), Metaheuristics- *Advances and Trends in Local Search Paradigms for Optimization*, Kluwer, Boston, 473-486.

[12] Kontoravdis, G. and Bard, J. F., 1995, A GRASP for the vehicle routing problem with time windows, *ORSA Journal on Computing*, 7, 10–23.

[13] Lenstra, J. and Rinnooy Kan, A., 1981, Complexity of vehicle routing and scheduling problems. *Networks*, 11, 221–227.

[14] Liu, F. and Shen, S., 1999, A route-neighborhood-based metaheuristic for vehicle routing problem with time windows. *European Journal of Operational Research*, 118, 485-504.

[15] Osman, I. H. and Kelly, J. P., 1997, Metaheuristics: An overview. In: Osman, I.H., Kelly, J.P. (Eds.), *Metaheuristics - Theory and Applications, Kluwer*, Boston, 1-21.

[16] Potvin, J. Y. and Bengio, S., 1996, The vehicle routing problem with time windows-Part II: Genetic search. *INFORMS Journal on Computing* 8, 165-172.

[17] Potvin, J. Y. and Kervahut, T., Garcia, B. L. and Rousseau, J. M., 1996, The vehicle routing problem with time windows - Part I: Tabu search. *INFORMS Journal on Computing*, 8, 158– 164.

[18] Schulze, J. and Fahle, T., 1999, Parallel algorithm for the vehicle routing problem with time window constraints. In: Beasley, J. E., Sharaiha, Y. M. (Eds.), Combinatorial optimization: Recent advances in theory and practice, *Special Issue of Annals of Operations Research*, 86, 585-607.

[19] Shaw, P., 1998, *Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems*. In: Maher, M., Puget, J.F.

(Eds.), Principles and Practice of Constraint Programming, Proceedings of the 4th Int. Conf., Pisa, Italy, Springer, Berlin, 417-431.

[20] Solomon, M. M. and Desrosiers, J., 1988, Time window constrained routing and scheduling problems. *Transportation Science*, 22, 1–13.

[21] Taillard, E. D., 1999, A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO*, 33 (1), 1-14.

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

[23] Tarantilis, C. D., Kiranoudis, C. T. and Vassiliadis, V. S., 1994, A backtracking adaptive threshold accepting meta-heuristic method for the vehicle routing problem. *Systems Analysis Modeling Simulation*, 42 (5), 631–664.

[24] Thangiah, S. R., Nygard, K. E. and Juell, P.L., 1991, *GIDEON: A Genetic Algorithm System for Vehicle Routing with Time Windows*. In: Proceedings of the 7th Conference on Artificial Intelligence for Applications, Miami, FL. *IEEE* Press, 322-328.

[25] Thangiah, S. R., Osman, I. H. and Sun, T., 1995, *Hybrid Genetic Algorithms, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows*. Technical Report UKC/OR94/4, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK.