Novelty Journals

ISSN 2394-9651

International Journal of Novel Research in Physics Chemistry & Mathematics
Vol. 3, Issue 1, pp: (64-74), Month: January-April 2016, Available at: www.noveltyjournals.com

An Integration between the Improved Ant Algorithm and the Simulated Annealing Algorithm to Contribute To Solve the Vehicle Routing Problem with Time Windows

[1]Dr. Mohammed Hassan Hassan, [2]Dr. Laina Makdyssiian, [3]Waseem Habib Bila

[1]Professor, Department of Mathematics, Faculty of Science, Tishreen University, Lattakia, Syria.

[2]Lecturer , Department of Communication, Faculty of Technology Engineering of Information and Communication, Tartous University, Syria .

[3]PhD student, Department of Mathematics, Faculty of Science, Tishreen University, Lattakia, Syria

Abstract: **In this research, we are studying the possibility of contribution in solving the Vehicle Routing Problem With Time Windows (VRPWTW), that is one of the optimization problems of the NP-hard type. This problem has attracted a lot of attention at the present time because of its Real Life applications. However, there is still no algorithm that provides us with the perfect solution to this problem because of the complexity of polynomial time. This means that the time of the solution to the Vehicle Routing Problem With Time Windows is growing steadily with the increase in the number of nodes. All the used algorithms have given solutions that are close to the optimal one. We'll show in our search Improved Ant Colony System algorithm (IACS) that is capable of searching multiple search areas simultaneously in the solution space is good in diversification, and Simulated Annealing algorithm (SA) is a local search technique that has been successfully applied to many NP-hard problems. In this research, we will present the In this research Hybrid algorithm (HA) Hybrid Algorithm provided (IACS-SA) that integrate between improved ant algorithm and Simulated Annealing algorithm . We will then compare the quality of the solution resulted from this hybrid approach with the results of well-known standard tests to determine the effectiveness of the presented approach.**

Keywords: **Vehicle Routing Problem With Time Windows, Simulated Annealing Algorithm, Improved Ant Colony System Algorithm, Hybrid Algorithm, 2-Opt Local Search, Nearest Neighbor Algorithm, Meta-Heuristics.**

1. INTRODUCTION

In the vehicle routing problem with time window (VRPTW), the objective is to minimize the cost travel. The Vehicle Routing Problem with Time Windows (VRPTW) is a well known optimization problem and it has received a lot of attention in operational research literature. In this problem, a fleet of vehicles must leave the depot, serve customer demands, and return to the depot, at a minimum cost, without violating the capacity of the vehicles as well as the time window specified by each customer.

The VRPTW has been dealt with various objectives. In the present article, it reduces the travelling distance and the number of vehicles simultaneously. The aim is to minimize the total travelling distance which is one of the most commonly found in literature, Dantzig [1] ,Solomon [2], Laporte, G [3] , Fisher [4] , Gillett [5] ,Golden [6], Lenstra [7] and , Lin [8] also adopted the same objective.

Given the complexity of the problem, its resolution using pure exact methods is often an extremely arduous task due to the large amount of computational time required. This fact has motivated the development of new heuristic algorithms for solving VRPTW. It is noteworthy to mention that such algorithms (hybrid algorithms) aim at finding near-optimal solutions using less computational effort.

The algorithm proposed in this article for solving VRPTW is hybrid method, (IACS-SA) that integrates between improved ant algorithm and Simulated Annealing algorithm, which periodically determines the best combination of routes generated during the execution of the algorithm.

The developed algorithm, that was tested on a set of 56 instances containing 100 customers was proposed by Solomon [2]. Moreover, These test-problems, including, the competition in 6 cases presented below, had been widely adopted in the literature and it is used to measure the efficiency of the exact, heuristic and hybrid methods.

Vehicle Routing Problem with time windows (VRPTW) is a variant of the Vehicle Routing Problem (VRP) with the additional time constraints. [1,2,3].This Problem is defined as follows: A number of vehicles is located at a single depot and must serve a number of geographically dispersed customers. Moreover, each vehicle has a given capacity and Each customer has a given demand and must be served within a specified time window $[e_i,l_i]$. The vehicle cannot arrive earlier than time e_i and no later than time l_i. The objective function is to minimize the total cost of travel and is expressed in the following equation:

$$\min F = \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}^{k} C_{ij}^{k} \qquad (1.1)$$

Where Decision Variable $\boldsymbol{x_{ij}^{k}}$ defines as follows:

$$\boldsymbol{x_{ij}^{k}} \triangleq \begin{cases} \mathbf{1} & \text{if vehicle k visits j after i, i } \neq \text{ j,} \\ \mathbf{0} & \text{otherwise} \end{cases} \qquad (1.2)$$

2. THE PROBLEM

The VRPTW, is represented by a graph $G = (V, E)$ is defined by a complete weighted and an undirected graph with a set of a vertices $V = \{v_0 . v_1, v_2, .., v_n\}$ and a set of edges $E = \{(i,j): i,j \in V . i \neq j$.The set $\{v_1, v_2, ..., v_n\}$ represents all customers and the $\{v_0\}$ represents a depot. Each customer i has an associated demand $q_i \geq 0$, a service time $s_i \geq 0$ and a time window $[e_i,l_i]$. For the depot $q_0 = 0$ and $s_0 = 0$. For each edge $(i,j) \in E$, and the travel cost c_{ij} is known. When Q is a capacity of identical vehicles. A matrix $C = (c_{ij})$ is defined on E and $c_{ij} = c_{ji}$ and $c_{ij} = \infty$. The cost function is $C: E \rightarrow Z^+$. The non-negative weights c_{ij} are associated with each arc (i,j) and represent the cost (distance, travel time or travel cost) between i and j. Each customer has a non-negative demand q_i and anon-negative service time t_i is given $(t_0 = 0 \ \& q_0 = 0)$. [2,3,4].

Fig 1: illustrates a graphical model of a simple VRPTW problem.

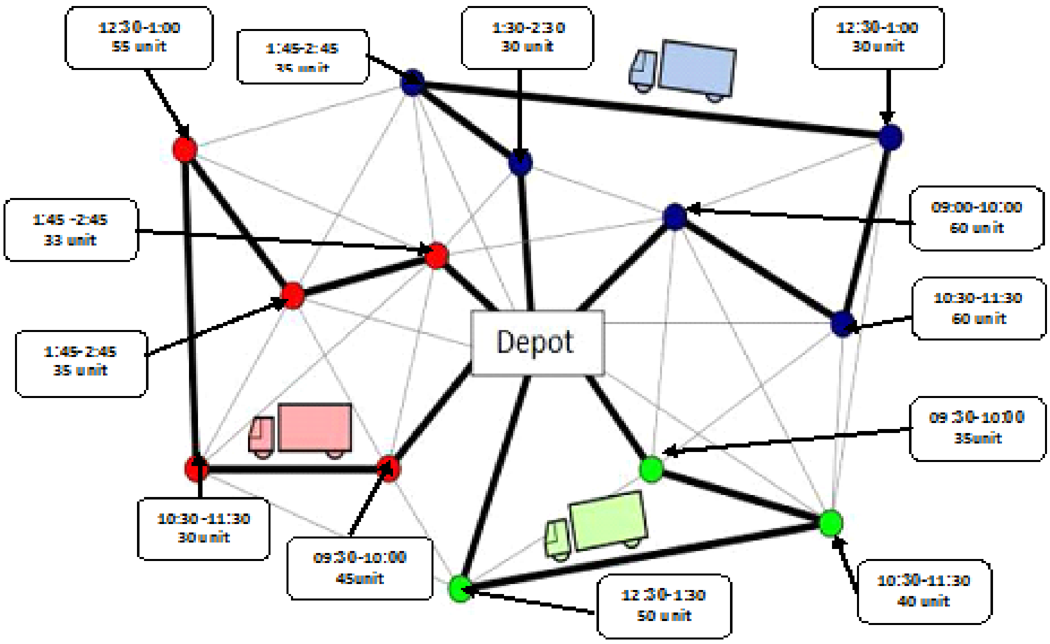

Figure: 1

3. THE ASSUMPTIONS

1- All routes start and end at the depot.

2- The total demand of any customer is not more than the capacity of the vehicle.

3- Each customer is serviced by only one vehicle.

4- Demand of each client is q_i is known.

5- All vehicles are homogeneous with known capacities Q are used.

4. OBJECTIVES

The aim of the VRPTW is to service all the C customers using the V vehicles while all the constraints are satisfied.

1- Minimize the total number of vehicles used to serve the customers.

2- Minimize the distance travelled by all the vehicles.

3- Schedule service-time and waiting-time without violating vehicles capacity and time windows constraints.

5. THE CONSTRAINTS

1- Each customer v_i is serviced only once by only one vehicle.

2- Serve every customer v_i within the time window $[e_i, l_i]$.

3- The total demand of any customer is not more than the capacity of the vehicle Q.

4- Every vehicle leaves the depot and returns to the depot v_0 .

5- The time window and capacity constraint are restricted to each vehicle.

ISSN 2394-9651
International Journal of Novel Research in Physics Chemistry & Mathematics
Vol. 3, Issue 1, pp: (64-74), Month: January-April 2016, Available at: www.noveltyjournals.com

6. THE PARAMETERS

K: is the total number of vehicles.

N: is the total number of customers.

C_{ij}: is the cost incurred on arc from node $i$to.

t_{ij} : is the travel time between node i and j , $(t_{ij} > 0)$.

m_i: is the demand at node.

q_k: is the capacity of vehicle k .

e_i:is the earliest arrival time at node i.

l_i: is the latest arrival time at node.

f_i:is the service time at node i.

r_k: is the maximum route time allowed for vehicle.

T_i: is the arrival time at node.

w_i: is the waiting time at node.

7. FORMULATION

Based on the above description. Laporte &Solomon et al. [2,3,4] .,the **VRPTW** can be formulated as follows:

$$\sum_{k=1}^{K}\sum_{j=1}^{N} x_{ij}^{k} \leq K , for i = 0 \quad (7.1)$$

$$\sum_{j=1}^{N} x_{ij}^{k} = 1 \; for i = 0 \; \& k_L \in \{1, \dots \dots, K\} \quad (7.2)$$

$$\sum_{j=1}^{N} x_{ji}^{k} = 1 \; for i = 0 \; \& k_L \in \{1, \dots \dots, K\} \quad (7.3)$$

$$\sum_{k=1}^{K}\sum_{j=0, j\neq i}^{N} x_{ij}^{k} = 1 , \; for i \in \{1, \dots .. N\} \quad (7.4)$$

$$\sum_{k=1}^{K}\sum_{i=0, j\neq i}^{N} x_{ij}^{k} = 1 , \; for j \in \{1, \dots .. N\} \quad (7.5)$$

$$\sum_{i=1}^{N} m_i \sum_{j=0, j\neq i}^{N} x_{ij}^{k} \leq q_i \; , for k_L \in \{1, \dots \dots, K\} \quad (7.6)$$

$$\sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N} x_{ij}^{k} \left(t_{ij} + f_i + W_i\right) \leq r_k , \; for k_L \in \{1, \dots .. K\} \quad (7.7)$$

$$T_0 = f_0 = W_0 = 0 \quad (7.8)$$

$$\sum_{k=1}^{K}\sum_{i=0, i\neq j}^{N} x_{ij}^{k} \left(T_i + t_{ij} + f_i + W_i\right) \leq T_j , \; for \; i \in \{1, \dots .. K\} \quad (7.9)$$

$$e_i \leq (T_i + W_i) \leq l_j \; , \; for \; i \in \{1, \dots .. N\} \quad (7.10)$$

International Journal of Novel Research in Physics Chemistry & Mathematics
Vol. 3, Issue 1, pp: (64-74), Month: January-April 2016, Available at: www.noveltyjournals.com

Constraint set (7.1) guarantees that the number of tours is k by selecting at most K outgoing arcs from the depot ($i = 0$).The constraint set (7.2) ensures that for each vehicle, there is exactly one outgoing arc selected from the depot.Similarly, the constraint set(7.3) ensures that for each vehicle, there is exactly one arc entering into the node with respect to depot ($i = 0$).These two constraint sets (constraint set (7.2) and constraint set (7.3)) jointly ensurethat a complete tour for each vehicle is ensured .

The constraint set (7.6) sees that for each vehicle, the total demand (load) allocated to it is less than or equal to its capacity. The constraint set (7.9) guarantees that the arrival time vehicle at the node j is less than the specified arrival time at that node.The constraint set (7.10) guarantees that the sum of the arrival time and the waiting time of each vehicle at each node i is more than equal to the earliest arrival time at that node and less than or equal to the latest arrival time at that node $i, i = 1, 2, 3, \cdots, N$.

Constraint sets(7.8 -7.10) define the time windows. These formulations completely specify the feasible solutions for the VRPTW. The constraint set (7.4) makes sure that from each node i only one arc for each vehicle emanates from it. The constraint set (7.5) ensures that for each node j, only one arc for each vehicle enters into it . These two constrains (constraint set 7.4 and constraint set 7.5) make sure that each vehicle visits each node only once.

8. METHODS OF SOLUTION

Three kinds of algorithms are used to resolve the VRPTW. Laporte & Solomon t al. [2,3,4] .

8.1 Exact Algorithms:

The most sophisticated exact algorithms for the VRPTW is that the algorithms are suitable for situations of a relatively a small scale, and cannot solve the problem of the cases of more than 100 customers in a reasonable period of time. Moreover, because VRPTW is of the NP-hard class, it is difficult to resolve even with the use of computers. Examples of Exact Algorithms are:

8.1.1 Branch and Cut.

8.1.2 Branch and Cut and Price.

8.2 Approximate Algorithms:

Most of the research has focused on the design of Classical Heuristics and Metaheuristics. Those two kinds of Heuristics are approximation algorithms that aim at finding good feasible solutions quickly with iterative improvements for the problems on a large scale and are divided into two main groups:

8.2.1 Classical Heuristics:

Although the heuristic approach does not guarantee optimality, heuristic methods often produce near-optimal solutions in a reasonable amount of computational time and it yields best results in practice. [5,6].

Since the overall search is impractical, and the extensive research is not practical, Classical Heuristics are used to accelerate the search, but it is not effective to escape from falling into Local Optimum. There is a big gap between the resulting solutions and the best known solutions. Examples of heuristics algorithms are:

1- local Search algorithm;

2 - The sweep algorithm.

8.2.2 Metaheuristics:

The metaheuristics can guide the search procedure to escape from local optima and jump to another solution space. Therefore, it can gain better solutions.

Over the past decade, a few new metaheuristics for VRPTW [4,5,10].were published which can improve the computational results of some benchmark problems. Examples of metaheuristics algorithms are:

1. Ant Colony Algorithm.

2. Genetic Algorithm.

8.3 Hybrid Algorithms (HA):

In this research, we try to provide a new algorithm to get better and more effective results. Bent et al. [8,9]. Many researchers have found recently that the employment of hybridization in the optimization problems could improve the quality of solutions that can be found in comparison with the heuristics and meta heuristics approach. In spite of the large and important differences between these algorithms, they share some of the items that have been ignored and left untapped. We will use the Hybrid Algorithms to focus on the strengths and compensate the weaknesses in the previous approximate algorithms. Therefore, the goal is to combine intelligently between the key elements of the competing methodologies to find a better solution.

9. PROCESSING THE PROBLEM

Since VRPTW is of NP-hard class, Lenstra & Rinnooy Kan. [6,7].We will use the hybrid Algorithms form the framework for Processing the VRPTW. We will display the Approximate Algorithms used to Process the problem and the presented Hybrid Algorithm.

9.1 Simulated Annealing Algorithm:

The main idea of a Simulated Annealing algorithm (SA) is to occasionally

Accept degraded solutions in the hope of escaping the current local optimum.

As a generic probabilistic method, simulated annealing has a capability to obtain the near optimum solution for problems with large search space. The method has been successfully implemented for various combinatorial problems. Simulated annealing method is inspired by the physical process of annealing where a material such as steel or glass is heated and then cooled By simulate the physical process; [11,12].

9.1.1 Steps of the Simulated Annealing Algorithm:

Input: problem.

Output: X_{best}, the best solution found so far by the algorithm;

1- Set barometers: T initial temperature, C cooling parameter, M maximum number of iterations. m maximum number of move operator .

2- Generate an initial solution x_0 using the nearest neighbor heuristic algorithm. set $= x_0$.

3- Apply the local search algorithm 2-opt the initial solution , $i = 1$, $j = 1$

4- Compute the objective function value of current solution $f\ (x)$.

5- a - If $i \leq M$ then apply the move operator local search algorithm 2-opt to the current solution to generate new solutions N, $i\ =\ i\ +\ 1$ and then move on to Step 5b, and otherwise ,go to step 6.

b- evaluate $\Delta E\ =\ f\ (N)\ -f\ (x)$ .if ΔE≤0 go to step 5; otherwise move go to Step 5 c .

c- Select a random variable $u \sim U\ (0,1)$, and if $u < p(\Delta E) = \exp(-\frac{\Delta E}{T})$ move to step 5 d, otherwise go to step 5a.

d- Accept the exchange, set $x\ =\ N$ and $f\ (x)\ =\ f\ (N)$, then go to step 5a .

6- If $i \leq M$ evaluate $T = r$ and $j = j + 1$, and then go to step 3; otherwise stop.

9.2 2-Opt Algorithm:

The local search simple algorithm was used to solve the Travelling Salesman Problem (TSP). In implementation, 2-opt local search procedure is used to obtain more improvement in the algorithm's performance [13]. This algorithm, which is shown in Figure 3, Show local search algorithm 2-opt, arc (3.6) and (0.4) and then reconnecting arcs (0,6), (3,4).

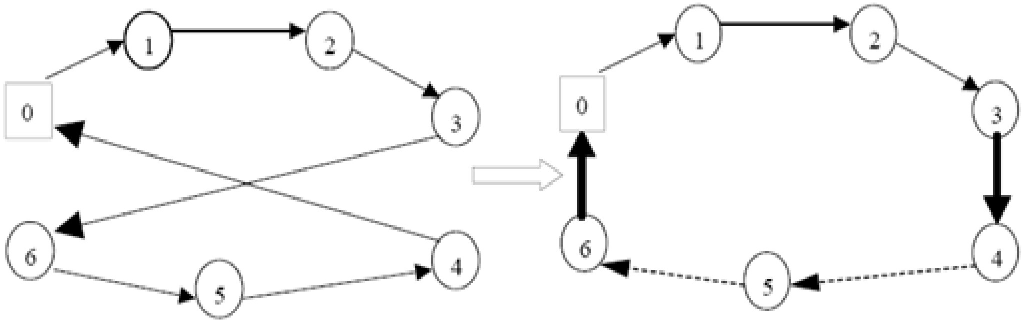

Figure 3: 2-opt Local Search.

Is based on omitting two arcs of the tour that is not adjacent and connecting them again by another method to find the optimal tour . It can be noted that there are several routes for connecting nodes and producing the tour again, and then for repeating this process by using a different set of other two arcs, with the problem's constraints being acceptable. So, this unique tour will be accepted only if, first, the above constraints are not violated, especially regarding each vehicle's capacity, while respecting the constraints of both the time windows and the capacities .

9.3 Nearest Neighbor heuristic Algorithm (N.N):

The nearest neighbor algorithm [7,8] is an approximation algorithm to solve the travelling salesman problem. The algorithm only gives a rough attempt to obtain a better tour. Here are the steps of the algorithm:

Step 1: Start with on a depot node as the beginning of a new route.

Step 2: Find out the nearest unvisited node P and add it tocurrent tour.

Step 3: Step 3. Set the current node to P.

Step 4: If all the nodes are visited, then terminate.

Step 5: repeat Step2 until the vehicle capacity or maximum service time constraint is violated; then, drop the last node from the current route, and go to Step 1.

9.4 Improved Ant Colony System (IACS):

The Improved Ant Colony System IACS was first proposed by Ting and Chen (2004). The IACS, which is based on the ACS proposed by Dorigo and Gambardella (1997). Zhang & Tang [9], includes three steps as follows:

9.4.1 Solution Construction:

In the original ACS, each ant moves from present node i to the next node v according to therule given by (1):

$$j = \begin{cases} argmax_{j \in U}\{[\tau_{iu}]^{\alpha}[\eta_{iu}]^{\beta}\} if q < q_0 \\ J otherwise \end{cases} \quad (9.4.1)$$

J is a random variable generated according to the following function of distribution :

$$J: p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in J_k(i)}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} & if\ j \in J_k(i) \\ 0 & otherwise \end{cases} \quad (9.4.2)$$

Where U is the set of nodes which are not visited yet, τ_{ij} is the pheromone of edge (i, j), η_{ij} denotes the savings of combining nodes i and j on one tour as opposed to serving them on two different tours. Thus, the η_{ij} is calculated as follows:

$$\eta_{ij} = d_{i0} + d_{0j} - d_{ij} \quad (9.4.3)$$

Where d_{ij} denotes the distance between nodes i and j, and node 0 is the depot; and α, β are theparameters that determine the relative influence of pheromone versus distance $(\beta, \alpha > 0)$. q is a random variable that follows a uniform law on [0,1]&q_0: $(0 \leq q_0 \leq 1)$ is a parameter determines the relative importance of exploitation (arg) versus exploration (j).the next node is chosen according to (9.4.2).

9.4.2 Local Pheromone Update:

When an ant k goes to client, we use the following formula to update locally the track of pheromone on the arc (i, j) :

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\tau_0 \qquad (9.4.4)$$

where τ_0: $(\tau_0 > 0)$ is initial pheromone as defined in eq.

$$\tau_0 = \frac{1}{n l_{nn}} \qquad (9.4.5)$$

Where n is the number of nodes and l_{nn} is the tour length produced by the Nearest Neighbor(NN) heuristic.

9.4.3 global pheromone Update:

After improving the 2^{nd} solution by local search, our global updating rule is applied to the first two solutions. The rule is described as follow:

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\sum_{k=1}^{m}\Delta\tau_{ij}^{K} \qquad (9.4.6)$$

$$\text{Where: } \Delta\tau_{ij}^{K} = \begin{cases} \frac{L_{u+1}-L_k}{L_{u+1}} & if\ (i,j) \in \text{tour obtained by ant k} \\ 0 & \text{otherwise} \end{cases} \qquad (9.4.7)$$

where L_k is the length of tour obtained by ant k, u is the number of solutions whose global pheromone will be updated and equals to 2 in this Search.

9.4.4 The procedures of our IACS are described as follows:

Step1: Set parameters.

Step2: Generate an initial solution using Nearest Neighbor heuristic.

Step3: Apply the local search (2-opt) to the initial solution and let it to be the solution1 of population. g = 1, h = 2.

Step4: Construct solutions base on the route construction rule and progress local pheromone update by Eq. (4). h = h + 1.

Step5: If h > m, then h = 2 and go to Step 6. Otherwise, go to Step 4.

Step6: Sort the solutions 2~m in ascending order and apply local search (2-opt) to the 2^{nd} solution.

Step7: Apply the global pheromone update rule by Eq. (9.4.6).

Step8: Record the best solution so far and let it to be the solution 1 in the next generation. g= g + 1.

Step9: If the stopping criterion (maximum number of generations, G, in this paper) is met, then stop and output the best solution. Otherwise, go to Step 4.

10. THE PROPOSED HYBRID ALGORITHM

In this research, we will present the Hybrid algorithm (IACS- SA) that integrates the improved ant algorithm (IACS) with Simulated Annealing

Algorithm (SA). while construction algorithms produce themselves a feasible solution, improvement algorithms improve the solutions with having that feasible solution. Here, a construction algorithm, namely SA, and two improvement algorithms, namely ACS and 2-opt local search, are used. In this algorithm, the ACS is applied for improving every route of the vehicle; however, the nodes of each vehicle should be unchanged. On the other hand, the 2-opt local search is used for changing nodes and improving each vehicle.

International Journal of Novel Research in Physics Chemistry & Mathematics
Vol. 3, Issue 1, pp: (64-74), Month: January-April 2016, Available at: www.noveltyjournals.com

10.1 Steps of the proposed hybrid algorithm (SA+ACS)

1- n is number of nodes;

2- m is number of vehicles ;

3- s^* is the random solution ; // s^* is the best solution found //

4- $f^* = \infty$;

5- set parameters of the proposed algorithm ;

6- Initialize pheromone trails;

7- Use SA to order that should be visited by vehicles with respect to the depot and called it R.

8- gain all of the situations for R and set them to the matrix T .

9- for $i := 1$ to $length(T)$ do //main cycle//

10- begin

11- use SA for $T(i,:)$ to produce a solution called $T^*(i,:)$

12- for the rout of each vehicle in $T^*(i,:)$, execute the ACS;

13- if new solution is better than $T^*(i,:)$, replace it;

14- Apply $2 - Opt$ local search algorithm for $T^*(i,:)$

15- If f $(T^*(i,:)) < f^*$ then

16- begin

17- f^*:=f($T^*(i,:)$);

18- S^*= $T^*(i,:)$;

19- end //save the best so far solution//

20- end //main cycle//

21- Show S^* and f^*

22- End //procedure//

10.2 Experimentation and results:

These experimental results have been conducted, using C + + language, on a PC using a wizard corei3 and 2 GB of RAM, and some of the inputs and outputs standard that are known for VRPLIB[14].

Table 1.The computational results hybrid algorithm by comparing them with standard known results

Problem Type	Travel distance	No. of good solutions	No. of standard solutions	Time standard status(s)	algorithm provided(s)	RPD
R1	1203.56	4	12	639	440	2.46%
R2	932.23	4	11	722	435	0.79%
C1	828.76	0	9	435	240	0.16%
C2	590.49	0	8	431	360	0.00%
RC1	1365.35	3	8	586	400	1.36%
RC2	1079.81	3	8	662	380	0.28%
sum	56429	14	56	3475	2255	0.55%

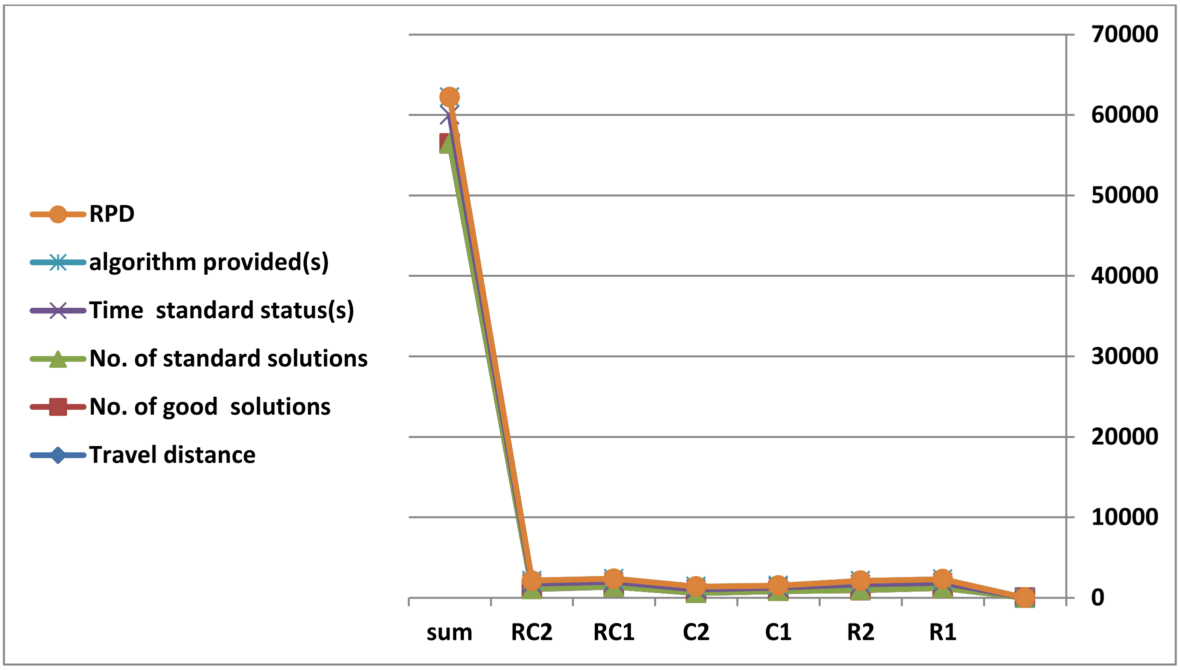

Figure 2: Computational results.

11. CONCLUSION

We have presented, a hybrid algorithm, combining ACS, Simulated Annealing and 2-opt local search, that was proposed for solving the Vehicle Routing Problem with Time Window. The hybrid algorithm (SA+ ACS) has more efficient results compared with the known benchmark problems results. It also gives better results, within a reasonable time, the heuristics algorithm and metaheuristics algorithm.

Its extension requires a lot of time, and the results showed that the efficiency of the proposed algorithm is the best hybrid where increased performance and reduced cost to solve the problem of directing the vehicle was a small gap between this algorithm and the results of a standard known. Moreover, Improving IACS-SA to reduce traveled distance and number of vehicles simultaneously is in progress. Thus, it is believed that the performance of IACS-SA is the best heuristics in solving VRPTW problems.

REFERENCES

[1] DANTZIG, G.B.; RAMSER, J. H., *"The Truck Dispatching Problem," Management Science,* Vol. 6, No. 1, 1959, pp. 79-89.

[2] SOLOMON, M.M. *Algorithms for the vehicle routing and scheduling problems with time window constrains*, Operational Research, Vol. 35, No. 2, 1987, 250-265.

[3] LAPORTE, G., "*The Vehicle Routing Problem: An Overview of Exact and Approximate of Operational Research,"* Vol. 59, No. 3, 1992, pp. 345-358.

[4] FISHER .M., *"Vehicle Routing. Handbooks of Operations Research and Management Science*," chap.1, 1995, pp.8:1-31**.**

[5] GILLETT, B.E.; L.R. MILLER, *"A Heuristic Algorithm for the Vehicle Dispatch Problem,*" Oper Res, 1974, pp.22:341–347 .

[6] GOLDEN, B.L; ASSAD,A.A., *"Vehicle Routing: Methods and Studies,"* eds. North Holland, Amsterdam,1988.

[7] LENSTRA,J.; RINNOOY KAN, A., *"Complexity of Vehicle Routing and Scheduling Problems,*" Networks, Vol.11, 1981, pp.221-227.

[8] LIN, S.W.; LEE, Z.J.; YING, K.C.; LEE, C.Y., *"Applying Hybrid Meta-Heuristics for Capacitated Vehicle Routing Problem,"* Expert Syst Appl, 2009, pp.36:1505–1512..

[9] ZHANG, X.; TANG, L., *"A new Hybrid Ant Colony Optimization Algorithm for the Vehicle Routing Problem,"* Pattern Recognit Lett, 2009, pp.30:850–855.

[10] DORIGO, M.; BIRATTARI, M.; STUZLE, T., *"Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique,"* IEEE Computational Intelligence Magazine, 2006, pp.139-147.

[11] KIRKPATRICK, S.;GELATT, C.D.; VECCHI, M.P. *Optimization by Simulated Annealing. Science*, vol 220, No. 4598, 1983. pp671-680.

[12] METROPOLIS, N., ROSENBLUTH, A.W., ROSENBLUTH, M.N., TELLER, A.H., TELLER, E. *Equation of State Calculation by Fast Computing Machines*. J. of Chem. Phys., 21, 1953.pp 1087-1091.

[13] http://en.wikipedia.org/wiki/2-opt .

[14] http://elib.zib.de/pub/mp-testdata/vehicle-rout/vrplib/VRPLIB.