Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management
Bali, Indonesia, January 7 – 9, 2014

A Comparative Study of Improved Simulation Annealing On Multi-Depot Vehicle Routing Problem with Time Windows

Subodha Dharmapriya
Department of Transport and Logistics Management, Faculty of Engineering
University of Moratuwa, Sri Lanka

Asela Kulathunga and Pradeep Jayathilake
Department of Production Engineering, Faculty of Engineering
University of Peradeniya, Sri Lanka

Abstract

Transportation, a vital logistics function, currently operating at a higher cost in industrial sector. Intelligent planning of transportation activities in distribution network with the assistance of Computational intelligence techniques is well understood a necessity. Here a Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW), which is NP-hard in nature, is considered. One of the meta-heuristics techniques: Simulated Annealing (SA) was tested with different modifications to derive at a high quality solution. Comparison has been carried out between adaptive and basic SA in order to show the importance of problem specific modifications of SA. Simulations were carried out on benchmark data and results revealed that improved SA out perform in the solution quality.

Keywords
Meta-heuristics, Computational Intelligence, NP-hard problem, Logistics

1. Introduction

Logistics cost is burning and emergent issue in all the fields. Ballou mentioned that logistics cost is average about 12% of the world's GDP. The major component of logistics is transportation, which occurs at different scales within the supply chain in the real world scenario. Therefore, an effective management and co-ordination of transportation in distribution of consumer products is vital for any kind of thriving supply chain.

Besides, it was shown that, electronically or mathematically generated optimal route leads to minimize travel distance (Laporte1995). Many well defined transportation problems found in the literature that have been formulated to optimize the transportation and distribution operations. The Travelling Salesman Problem (TSP) is the most primary, extensively studied problem in this area (Archetti et al. 2008). It is applied to find the shortest route to visit given sets of customers with many implied constraints and conditions. However, TSP is not applicable in certain instances where more than one vehicle is needed for the routing. Therefore, TSP has been branched off to different directions and also extended to Vehicle Routing Problem (VRP) in order to overcome such inefficiencies in TSP. VRP is further relaxed by adding conditions like Multiple Depot (MD), Time Window(TW), Split Delivery(SD) etc. Integration of above extensions would make the formulation more realistic. Hence, VRP has been extended with the conditions of multiple depot and time window in this study. The formulated problem is termed as Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW)

The approaches found in the literature are categorized into two basic categories namely *exact* and *heuristic* approaches (Eksioglu et al. 2009). The exact approaches can guarantee an optimal solution for small scale problems within reasonably lesser time (Kellehauge 2008). In contrast, the problems which are large in size or complex in nature need to get the aid of AI techniques. Considerable number of researchers in the Operations Research area has been turning towards heuristic, meta-heuristics to solve engineering management problem. Also these techniques have been successfully applied in NP-hard problems where the running time grows exponentially with the problem size (Kellehauge 2008).

Meta-Heuristics such as Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) are highly attractive for a complex problem as they are capable of generating high quality solution within relatively short computational time (Zachariadis et al. 2007). This study investigated the performance of improved SA on above defined transportation problem.

The significance of this study is, addressing a transportation problem which is closer to real life scenario and investigating the directions to improve SA heuristics. The performance of improved SA heuristic is investigated with comparison to conventional SA on MDVRPTW problem. The structure of the paper is as follows, section II describes the problem formulation, section III brings the methodology and section IV and V present simulations results and conclusion respectively.

2. Problem Formulation

The mathematical formulation of the problem is presented in this section. Definitions of variable and notations used in this study are described below.

Customers: A distribution network consists of *n* number of customers where$i \in C$ and i = 1,2,………n, residing at *n* different locations, catered by distributed warehouses (WH_j) where j = 1…….m and m < n. A customer is defined with a demand (q_i) where $q_i>0$ having the time window i.e an interval $[a_i,b_i] \in R$ where $a_i$and b_i are the earliest and latest times for start servicing the customer . Every pair of customers (i,j) where $i,j \in C$ and $i \neq j$ is associated with a traveling distance d_{ij} and traveling time t_{ij}.

Vehicle: A fleet of homogeneous vehicle V_n where n=1,2…..p with the capacity of V_c scattered throughout the network, allowing to start from any depot and reaching the place where it is requested. In this study, the demand of a customer may be satisfied by more than one vehicle when the demand of the customer exceeds the vehicle capacity or instances of delivering single demand by more than one vehicle which is known as split delivery.

$$X_{ijk} = \begin{cases} 1 \text{ ; If the vehicle k travel to demand point j directly from demand point i} \\ 0 \text{ ; Otherwise} \end{cases}$$

$$Y_{ik} = \begin{cases} 1 \text{ ; If the demand point i is visited by vehicle k} \\ 0 \text{ ; Otherwise} \end{cases}$$

$$\delta_{imk} = \begin{cases} 1 \text{ ; If vehiclektravelstodemandpointidirectlyfromwarehousem} \\ 0 \text{ ; Otherwise} \end{cases}$$

u_{ik} = dummycontinuousvariablesforsubtourelimination constraints

p_{ik}: Delivery amount at demand point i by vehicle k
q_i: Demand at customer i
a_i: Earliest arrival time of customer i
l_i: Latest arrival time of customer i
d_{ij} : Distance between customer i and j
e_{im}: Distance between customer i and warehouse m
c : Cost factor
V_c : Vehicle capacity

Minimize Z(d) =

$$C\left[\sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij}\sum_{k=1}^{u} X_{ijk} + \sum_{i=0}^{N}\sum_{m=0}^{P} e_{im}\sum_{k=1}^{u}\delta_{imk}\right] \quad (1)$$

$$\sum_{k=1}^{u} p_{ik} = q_i \quad (2)$$

$$p_{ik} \le q_i y_{ik} \quad (3)$$

$$a_j^k \le a_i^k + t_{ij} \le l_j^k \quad (4)$$

$$a_j^k \le l_i^k + t_{ij} \le l_j^k \quad (5)$$

$$\sum_{i=1}^{n} y_{ik} q_i \le v_c \quad (6)$$

$$\sum_{j=0}^{N} X_{0jk} = 1 \quad (7)$$

$$\sum_{J=0}^{N} X_{ijk} = \sum_{J=0}^{N} X_{jik} = Y_{ik} \ \forall i, \forall k \quad (8)$$

$$u_{ik} - u_{jk} + N.X_{ijk} \le N - 1 \ \forall i, j, k \quad (9)$$

Equation (1) illustrates the objective function for minimizing the cost of total traveled distance. Several constraints were identified in the system which is presented in equation 2 to equation 6 accordingly. Equation (2) guarantees that the demand of each point is totally satisfied by single or multiple vehicles. Equation (3) ensures that a vehicle only serves the demand point which is visited by the vehicle and also it is less than or equal to demand of the respective customer. Equation (4) and (5) refers the time window of a particular customer and (6) ensures that total distribution of the vehicle does not exceed the defined capacity of the vehicle. Equation (7) and (8) state that each vehicle leaves the depot, after arriving at the customer the vehicle leaves again and it will finally return to the depot. Equation (9) is the sub tour elimination constraint.

3. Methodology

The proposed method was folded into two main stages as shown in Figure 1. Initialization is performed by a deterministic approach and optimization is done by both basic and improved SA meta-heuristics. In addition, a comparison has been carried out to show the avenues of improving the SA regardless of its stochastic nature.

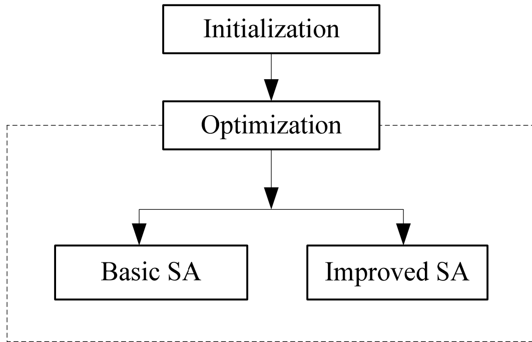

Figure 1. Structure of the methodology

Starting with a good initial solution is important in deriving a high quality solution in any of the heuristics and meta-heuristics (Eglese 1990, Elbeltagi et al. 2005). Sets of initial routes are generated at the initial stage considering the vehicle capacity, time window constraints and distance of a particular customer. The outline of the initialization is illustrated in the figure 2. The customer is considered as a task in this study and each and every task in the distribution network should have information on location (coordinates), due date and demand in advance. Based on the location (coordinates), the distance between each demand point is calculated taking Euclidean distance. All received tasks have to be processed in such a way that tasks are separated according to the due date and then sorted by the earliest arrival time of the due time interval. Here, the boundaries of the time interval are named as the earliest and the latest arrival time. The selection of warehouse (depot) is done considering the closeness to the first non-assigned task of the prepared task list and the availability of the goods in the warehouse. If the vehicle is capable of loading more tasks, they are selected based on the distance and the availability of vehicle space. However, the first feasible task of the previous task should satisfy the time window or else algorithms consider the next feasible task. This will be repeated until the vehicle is fully loaded with tasks and continue until all the tasks are assigned to the all available vehicles.

Step 1: Process all arrived tasks
1. Separate tasks daily
2. Sort them by earliest arrival time
Step 2: Start from the first task of the sorted task list. (i=1)
Step 3: Find the tasks which exceed the vehicle capacity. Those deliveries are to be split.
Step 4: Calculate the remaining space after selecting the first task
Step 5: Do while until the remaining space >0
Select the tasks compatible with the remaining space
Find the shortest distance from the i^{th} task to all feasible tasks
Select the closest task
Do while until selecting the feasible task
If the tasks satisfy the time frame; select that tasks and set i(i = i+1)
else go to the next closest task and set i (i = i+1)
End
End
End
Step 6: Terminate the program after inserting all tasks to routes

Figure 2. Outline of the initial solution

3.1 Optimization

Solution obtained at the initial stage is optimized by SA meta-heuristic which is well known to be a stochastic algorithm. SA was introduced by Metropolis et al. in 1953 which has been effectively applied in number of combinatorial optimization problems such as computer design, image processing, molecular physics, job shop scheduling etc. (Eglese1990). Koulamas et al. (1994) have concluded that SA can be applied in both traditional (flowshop, jobshop, TSP) and non-traditional (graph coloring, number partioning) OR areas. Though SA is good in NP-hard problem, there are some inefficiencies which make long running time, trapping at local minimum and some other drawbacks. Comparatively a very few studies were found in the literature which worked on analyzing the physical analogy of SA. Most of the studies try to apply conventional SA in various applications (Koulamas et al. 1994). Eglese (1990), Azizi and Zolfaghari (2004) and VasanandRaju (2009) presented some theoretical perspectives to enhance the efficiency of SA.

The initial temperature, cooling rate and stopping criterion are considered to be the main parameters of SA where it is termed as the annealing or cooling schedule (Eglese1990). The selection of above parameters makes a significant influence on the performance of the algorithm. For instance, at higher temperature uphill moves are accepted whereas at lower temperature the most of uphill moves will be rejected (Eglese 1990, VasanandRaju 2009).

Though the SA is good in finding high quality solution reasonably lesser time than the exact algorithm, still room exists to improve the solution by investigating the improvements of parameters of SA. In order to achieve this objective, experiments were carried out on both conventional SA and the improved SA on MDVRPTW problem.

- Adaptive SA
  - Start with a more reliable initial point

Similar to other heuristics, conventional SA starts with a random initial solution and ending up with different solutions which may not be the global optimum but which is one of the local optimums. Moreover, in order to obtain a high quality results, it is required to start with a good initial solution, which makes the probability of finding the global optimum higher (Eglese 1990, Elbeltagi 2005). Therefore, this study derived the initial solution by a deterministic approach as mentioned in the initialization, which is one of the reliable points to start the search.

  - Alternative acceptance probabilities

$$\exp\text{ (-delta/T)} \tag{10}$$

$$\exp -(new\ solution - current\ solution)/T \quad \geq rand\ [0,1] \tag{11}$$

$$\exp -(new\ solution - current\ solution)/\log(T) \quad \geq rand\ [0,1] \tag{12}$$

Acceptance of the worse solution generated by the neighborhood search determines based on the Metropolis Acceptance Criteria. Metropolis acceptance criteria is derived from the Boltzmann function which is given in equation (10). The general Metropolis Acceptance Criterion is given in equation (11) and it has been modified in this study due to many reasons. Some of the previous researchers also tried to change the acceptance criterion with some theoretical study (Anily and Federgruen 1987, Falgle and Sehrader 1988, Romeo and Vincentelli 1985). Johnson et al (1987) stated that in many applications of SA, the value of exp (-delta/T) is an affecting factor for run time. Therefore, they proposed to approximate the exponential function by a simple table lookup scheme.

According to the general acceptance criteria, it is obvious that at high temperature accepting worse solution goes high whereas at lower temperature uphill moves are limited. By accepting highly deviated solution at high temperature, the descent direction of searching may change drastically. Ultimately it makes a considerable gap between optimal and the current solution. Therefore, the standard acceptance function has been modified in this study as shown in Singh and Deshpande (1993). The logarithm of temperature makes the denominator lower and therefore, the value of Boltzmann function becomes a larger negative value. According to the exponential curve in Figure 3, exponential value is closer to zero. Therefore, the probability of accepting the worse solution goes down.

This modified acceptance criterion has been used in high temperature range whereas at low temperature range the general acceptance function is applied.

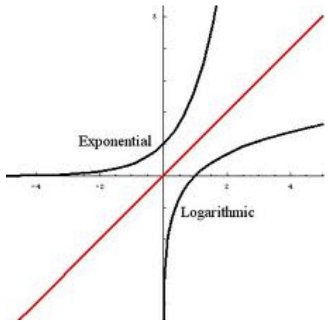

Figure 3. Logarithm and Exponential

- Effective cooling rate for geometric cooling schedule

T(i+1) = alpha(α) * T(i) (13)

The temperature at the running conditions and the energy difference decide the value of Boltzmann function. The temperature at each iteration is determined by the cooling function or annealing schedule. In the basic SA, the cooling function used is termed as geometric cooling function which is denoted in equation (13) where alpha is a constant which refers to the cooling rate. T (i) is the temperature at i^{th} succeeded step and T (i+1) is the defined temperature of the succeeded step (i+1). Succeeded step is defined when the movement which turns out better than the current solution or the worse solution is accepted based on Metropolis Acceptance Criterion. Number of simulations were carried out to see the significance of cooling rate and the appropriate cooling rate for distinct problem sizes. The cooling rate (α) is varied from 0.99 to 0.80 as it was said to be the effective range Eglese1990, Tsang and Wiese 2007. Significance of each value of the cooling rate was examined with respect to the solution quality and the elapsed time of the algorithm. Simulations relevant to the above studies are presented in the simulation results section.

4. Simulation Results

Simulations were carried out on benchmark data published in [18] on MDVRPTW problem. The data set is basically used to investigate the possible improvements of SA in order to derive at a high quality solution. As clearly explained under the methodology, the significance of appropriate cooling rate is described in this section. Experiments were carried out to see the effect of cooling rate on the elapsed time and quality of the solution as well.

4.1 Quality of the solution

The cooling rate acts as one of the deterministic criteria which decide number of iterations where it makes an influence on the quality of the solution. This can be clearly seen by observing the variation of the final solution obtained at different cooling rate as plotted in **Error! Reference source not found.**ure 4. Those results were taken by conducting numbers of simulations for different problem sizes. According to Figure 4 information, for smaller size problems (i.e problem size below 100 customers) the variation of best solution with cooling rate is fairly constant whereas significance drop can be seen in large scale problem when the cooling rate goes up.

Generally, lower cooling rate makes higher decrement in temperature which leads to lower number of iterations at a single run and also lower elapsed time. In contrast, higher cooling rate makes lower decrement in temperature which leads to higher number of iterations and elapsed time as well. The number of combinations in the large scale problem is higher than the small size problems. Therefore, for smaller size problems, it is better that the cooling rate is a lower value which makes higher decrement in the temperature. In contrast, for large scale problem, it is better to assign a higher value for cooling rate. Hence it will reduce the decrement in temperature which helps to increase the number of runs. This will helpful for large size problems to check relatively large number of alternative solutions which is ultimately beneficial to derive at a robust solution.

4.2 Elapsed time

As Figure 5 indicates, elapsed time of the algorithm is increased exponentially with the cooling rate. When the problem size increases, elapsed time goes up following the same pattern. However, it is not advisable to make a decision of setting up the cooling rate solely depending on the elapsed time. This is one of the simulations which highlight the significance of cooling rate for the overall problem in terms of time.

However, it is prudent to make decisions incorporating the elapsed time and solution quality simultaneously. It can be concluded by observing Figure 4 and Figure 5 which contains the information on the solution in terms of elapsed time and the quality, for large size problems (size above 100) cooling rate better to be higher value (0.98) whereas for small scale problem cooling rate can be lower one (0.96).

4.3 Effect of acceptance criterion and cooling rate

The behavior of the acceptance criterion is investigated and presented in this section. Fig. 6 shows the variation of the accepted delta value (difference in distance) at each temperature under the general acceptance criterion as in equation (11). It is noticeable that the value of these accepted solutions varies in between 0 – 600 where at higher temperatures, highly deviated worse candidate solutions are accepted than at the lower temperature. At lower temperatures it can be seen that small perturbations are accepted. Though the higher cooling rate is applicable for large size problems in terms of steps within stopping criteria, the quality of the final solution may be low due to the acceptance of highly deviated solutions at higher temperatures. Thus the general acceptance criterion was modified in order to overcome some of these issues.

Therefore, necessity arises to take remedies to overcome these inefficiencies and to make the search more robust. In order to reduce the steps at higher temperatures, the rate of reducing the temperature makes higher till an average temperature. Conversely, to make steps increase at lower temperatures, cooling function has been changed with the objective of increasing the number of steps. At higher temperature range, the reduction in temperature makes high by applying the cooling rate as in equation (10) and acceptance criterion is used as in equation (13).At lower temperature in order to have more steps, the cooling rates function is defined as $T(i+1) = T(i)/(1+0.0002*T(i))$ and general acceptance criteria (12) is used.

The effect of the proposed modifications can be clearly seen through Figure 6 – Figure 9 separately. Figure 6 and Figure 7 shows the algorithmic behavior with general and modified cooling function respectively. It can be clearly seen the intensity of the graph at lower temperature is higher in Figure 7 with compared to Figure 6. It reveals that the modified cooling rate helps to increase the number of perturbations. Therefore, it can be concluded that the modifications done on the cooling function could reduce the frequency of accepting worse candidate solution at higher temperature whereas increase at lower temperature range.

Figure 8 shows the algorithmic behavior with the modified acceptance criteria and general cooling rate. It clearly shows the reduce of range of accepted solution, it goes down to 0 – 40. Similarly, algorithmic variation with modified acceptance criterion and modified cooling function is shown in Fig. A significance difference can be seen in Figure 8 and Figure 9 with respect to the intensity of the graph. For the adaptive SA, conditions used in Figure 9 was applied.

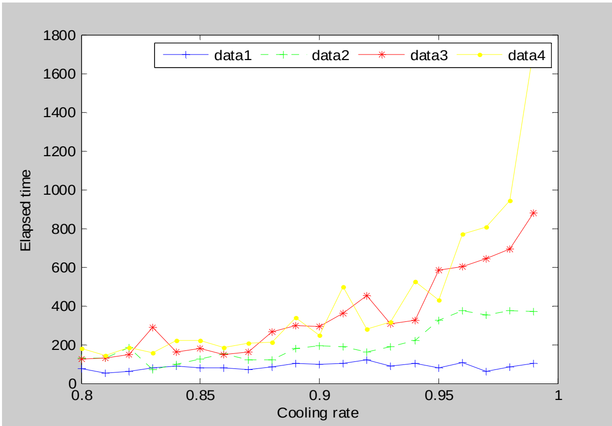

Figure 4: Variation of elapsed time with cooling rate

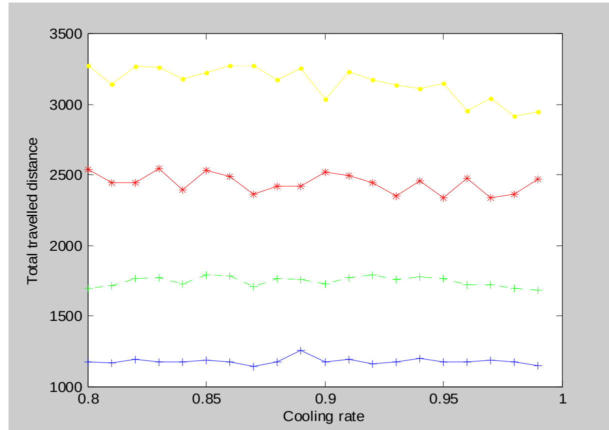

Figure 5: Variation of final solution with cooling rate

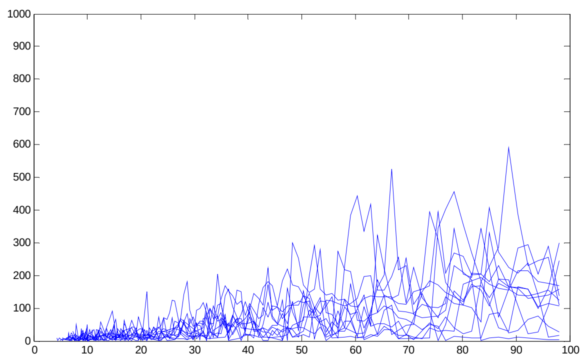

Figure 6: Algorithmic behavior with general cooling function

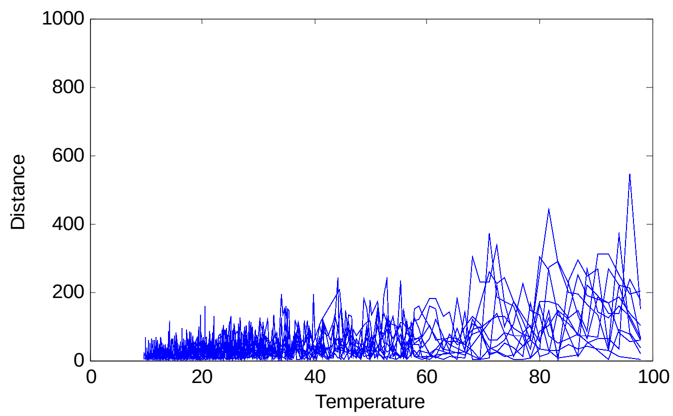

Figure 7: Algorithmic behavior with modified cooling rate

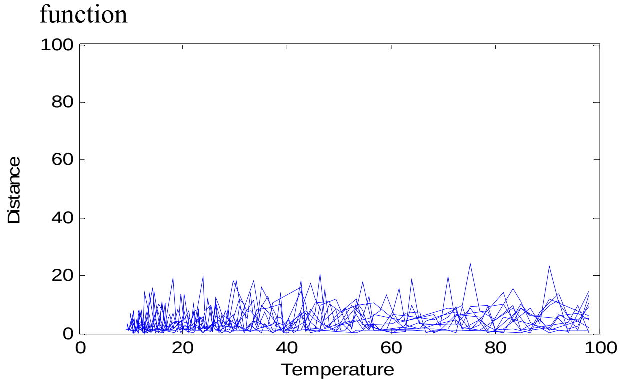

Figure 8: Algorithmic behavior with modified acceptance cooling rate and modified acceptance criterion

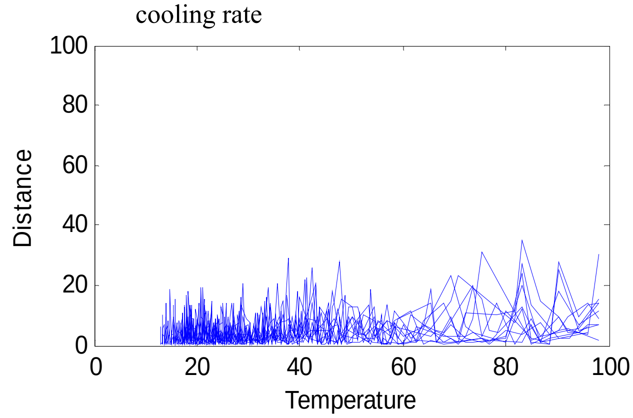

Figure 9: Algorithmic behavior with modified criterion and general cooling rate

Simulations results presented in Table 1 is based on the condition applied in Figure 6 and Figure 9 respectively. Algorithm was run on the bench mark data proposed by Cordeau et al. (2001). Table 1 shows the summary of the results, the shown value is the best value derived after 10 runs of the algorithm. By comparing the values in table 1, it can be concluded that adptive SA perform well in deriving high quality solution than the conventional SA.

Table 1: Comparison between performance of basic SA heuristics and improved SA heuristics

Problem size	Travelled distance with basic SA	Travelled distance with improved SA
48	1248	1230
96	1794	1137
144	2735	2735
192	3421	2906
240	3592	3255
288	4287	3955

5. Conclusion

The distribution network considered here is composed of multiple depots and customers with defined time windows. SA has been used for the simulations with some proposed changes in cooling schedule and metropolis acceptance criterion. It was found that, for smaller problems the cooling rate is sufficient to be a lower value (90-96 range), which will help to save time while giving good results. In order to limit the higher perturbations and number of decrement steps at higher temperature, modified acceptance criterion and general cooling schedule has been used. In contrast, for lower temperature range general acceptance criterion and modified cooling schedule has been used. With the integration of problem specific modifications and appropriate condition at each stage, adaptive SA could derive high quality solution than basic SA.

References

R.H. Ballou, "Business Logistics Management", 4th ed., Prentice Hall, pp.12.

G.O. Laporte, "Routing Problems:bibilography", Annals of operations research Eason, Vol.61, pp.227-262, 1995.

C. Archetti, W. Savelsbergh, G. Speranza, "To split or not to split: that is the question", Transport Research Part E, Vol.44, pp.114-123, 2008.

B. Kellehauge, "Formulations and exact algorithms for the vehicle routing problem with time windows". Computers and Operations Research , Vol.35, pp. 2307-2330, 2008.

C., Koulamas, S.Antony and R. Jaen, "A Survey of Simulated Annealing Applications to Operations Research Problems", Omega, Int. J. Mgmt Sci , Vol.22,pp. 41- 56, 1994.

E.Zachariadis, C. Tarantills, and C. Kiranoudis, "A hybrid meta-heuritics algorithm for the vehicle routing problem with simultaneous delivery and pick-up service", Expert Systems with Applications, 2007.

R. Eglese, "Simulated Annealing: A tool for operations research". European Journal of Operations Reesearch , Vol.46, pp.271-281,1990.

S.Anily, and A. Federgruen, "Simulated annealing method with general acceptance probabilities", Journal of Applied Probability , Vol.24, pp.657-667, 1987.

U.Falgle and R.Sehrader, " On the convergence of stationary sidtribution in simulated annealing algorithms", Information Processing Letters,Vol. 27, pp.189-194,1988.

F.Romeo and A.Sangiovanni-Vincentelli, "Probability hill climbing algorithms: properties and applications" *Proc.Chapel Hill Conference on VLSI*, pp. 393-417,1985.

D.Johnson, C. Aragon, L. McGeogh and C.Schevon, "Optimization by simulated annealing: An experimental evaluation", Part I. *AT & T Bell Laboratories.* Murray Hill,1987.

G.Singh and K. Deshpande. "On fast load partitioning by simulated annealing and heuristic algorithms for general class of problems". Advances in Engineering Software , Vol.16, pp.23-29,1993

Eksioglu, B., Vural, A., & Reisman, A. ,. A vehicle routing problem: A texonomic review. *Computer & Industrial Engineering, 57*, 1472-1483, 2009

N. Azizi, S. Zolfaghari, "Adaptive temperature control for simulated aneealing: a comparative study", *Computers & Operations Research 31*, pp. 2439-2451, 2004.

A. Vasan and K.S.Raju, "Comparative analysis of Simulated Annealing, Simulated Quencing and Genetic Algorithms for optimal reservoir operation", *Applied Soft Computing 9*, pp. 274-281, 2009.

E. Elbeltagi, T. Hegazy and D. Grierson, "Comparison among five evolutionary-based optimization algorithms", *Advaced Engineering Informatics 19*, pp. 43-53, 2005.

H.H.Tsang, K.C. Wiese, “A study of RNA secondary structure prediction using different annealing schedules”, IEEE symposium on computational intelligence in bioinformatics and compuutational biology, Surrey, 2007.
Cordeau, J., Laporte, G. and Mercier, A., 'A unified tabu search heuristic for vehicle routing problems with time windows', *Journal of the Operational Research Society*, vol. 10, p. 928–936,2001.

Biography

U.S.S. Dharmapriya is a postgraduate M. Phil student graduated from Faculty of Engineering, University of Peradeniya in 2004.

A.K. Kulatunga is a senior lecturer at Department of Production Engineering, University of Peradeniya.

P. Jayawardhana is a postgraduate student graduated from Faculty of Engineering, University of Peradeniya.