Journal of Theoretical and Applied Information Technology
10th March 2015. Vol.73 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE EFFECT OF HYBRIDIZING LOCAL SEARCH ALGORITHMS WITH HARMONY SEARCH FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

[1]ESAM TAHA YASSEN, [2]MASRI AYOB and [3]MOHD ZAKREE AHMAD NAZRI
[1,2,3] Center for Artificial Intelligence Technology, Faculty of Information Technology and Science, University Kebangsaan Malaysia, Bangi, Selangor, Malaysia
[1] Department of Computer Science, Computers College, University of Anbar, Iraq.
E-mail: e_t_972@yahoo.com, masri@ftsm.ukm.my, zakree@ukm.edu.my

ABSTRACT

Harmony search algorithm is relatively a recent nature inspired algorithm that mimics the musical improvisation process in seeking agreeable harmony. It has been used to solve various optimization problems and shown to produce good results for many problems. However, harmony search algorithm has slow convergence that might decrease its efficiency in producing good quality solutions for constrained optimization problems. In order to overcome this shortcoming, we propose a hybrid harmony search algorithm that integrates a local search algorithm and harmony search algorithm to increase its exploitation process and to further improve the generated solution. Three well-known local search algorithms: hill climbing, simulated annealing, and reactive tabu search are hybridized with the harmony search algorithm. The proposed algorithm is tested on the standard public Solomon's vehicle routing problem with time windows benchmark set. The computational experiment shows that each of the three hybrid algorithm produced good quality solution for certain instances only. However, all of them outperformed standard harmony search algorithm. Thus, it can be concluded that the integration of the local search with harmony search algorithm does improve its capability in producing good quality solutions.

Keywords: *Vehicle Routing Problem; Meta-Heuristics; Hybrid Algorithm; Hill Climbing; Simulated Annealing; Reactive Tabu Search; Harmony Search Algorithm.*

1. INTRODUCTION

The vehicle routing problem (VRP) is one of the most popular problems in the transportation and distribution systems [1]. Since it has been introduced by Dantzi and Ramser [2] in 1959, many variants of VRP were later introduced such as the capacitated vehicle routing problem, stochastic vehicle routing and the vehicle routing problem with time windows (VRPTW). Among the variants, VRPTW is most widely studied as it is closer to the real world situation [3] than other problems in VRP.

In VRPTW, given a set of geographically distributed customers and a set of vehicle with equal capacity, the objective is to find the minimum cost routes to serve all customers. The VRPTW is an NP-hard combinatorial optimization problem, and therefore exact methods can be used to solve problem with small size only [4, 5]. Consequently, meta-heuristic algorithms are usually used to solve the VRPTW, as they generate a reasonably good quality solution in a reasonable amount of time but there is no guarantee to obtaining an optimal solution [5, 6]. Examples of meta-heuristic algorithms that have been proposed to solve VRPTW are hybrid methods [7], [8], [9], [10], [11], greedy randomized adaptive search procedure [12], simulated annealing [13], tabu search [14], genetic algorithm [15], scatter search [16] and particle swarm optimization [4]. More details about VRPTW can be found in [17].

Harmony search algorithm (HSA) [18] is a recent nature inspired algorithm originally inspired by the natural process of musical improvisation that searches for suitable musical notes. Like other population based algorithm, HSA maintains a population of solution, which represents the potential solutions for a given problem. HSA differs from other population based approaches where it generates new solution by considered all existing solutions, whereas genetic algorithm for example consider only two solutions [19]. HSA has been proven by many researchers as an effective technique for solving several real world optimization problems including university course timetabling [20], Sudoku puzzle solving [21], web page clustering [22], the scheduling of multiple

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

dam system [23] and many others. Unfortunately, like other population based algorithms, HSA suffers from the slow convergence problem, especially when dealing with constrained optimization problems such as the VRPTW.

Hybridizing population based method with the local search method is very common in the literature, especially in evolutionary algorithm research filed which usually known as memetic algorithms [6]. The main purpose of the hybridization is to complement their strength. This is because population based methods are good in exploration, whilst local search methods are good in exploitation [6]. Due to flexible structure and simplicity of HSA, researchers like [24], [25], [26], and [27], hybridized HSA with other meta-heuristics. This hybridization enhances the exploitation ability of HSA in order to improve the quality of generated solution. Therefore, in order to improve the search ability of HSA, we have hybridized HSA with local search algorithms. The question is "can we enhance the performance of HSA by hybridizing it with local search?" That is, "if the hybridized algorithm (HSA with local search) can take advantages of HSA to explore the search space and the local search algorithm to exploit the search space, then the hybrid algorithm will perform better than HSA and local search". In this work, the hill climbing (HC) simulated annealing (SA) and the reactive tabu search (RTS) algorithms are used as the local search algorithms. These three local search algorithms have been hybridized with the HSA separately and the resultants are three different hybrid algorithms denoted as HSA-HC, HSA-SA and HSA-RTS. The local search algorithm will further improve the generated solutions by the HSA.

Computational experiments are carried out on a Solomon's VRPTW benchmark set in order to examine the performance of the proposed hybrid algorithm and compare it with standard HSA (without the local search algorithms), the three local search algorithms and the state of the art.

2. PROBLEM DEFINITION AND NOTATION

The VRPTW is formally defined as follows: given a set of customer with specific demand and a set of vehicle with equal capacity, design a set of routes to serve all customers with minimum traveling distance [3, 28]. The designed routes should respect the following:

a. Each route must start and end at the depot and each customer must be served only once and during its time window.
b. If the vehicle arrived at the customer c_i earlier than its start time window(e_i), the vehicle must wait and should serve the customer c_i during its time windows only.
c. The vehicle cannot serve customer c_i if it arrived after the end time window (l_i) of that customer (i.e. the customer c_i must be served by other vehicle).
d. The total demands of all customers in each route must be less than or equal to the vehicle capacity Q.

The main objective is to minimize the total traveling distance for all vehicles as much as possible. Please note that the solution in VRPTW contain a set of routes [4]. Consider the following variable:

v	*number of serving vehicles.*
n	*number of customer nodes (excluding the depot node).*
q_i	*demand of costumer* c_i.
Q_k	*capacity of the* k^{th} *vehicle.*
s_i	*service duration of costumer* c_i.
t_i	*the arrival time at customer* c_i.
t_{ij}	*travelling time from customer* c_i *to customer* c_j.
W_i	*waiting time at customer* c_i.
ei	*start time of time window for customer ci.*
l_i	*end time of time window for customer* c_i.

The VRPTW can be described mathematically as follows [4]:

$$x_{ij}^{k} = \begin{cases} 1 & \text{if vehicle } k \text{ travelled directly from customer } c_i \text{ to } c_j \\ 0 & \text{Otherwise} \end{cases}$$

$$y_{i}^{k} = \begin{cases} 1 & \text{if customer } c_i \text{ is served by vehicle } k \\ 0 & \text{Otherwise} \end{cases}$$

The quality of the S^{th} solution is measured as:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\min \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{v} t_{ij} \times x_{ij}^{k} \quad (1)$$

s.t.

$$\sum_{i=0}^{n} x_{ij}^{k} = y_{j}^{k} \quad \forall k = 1,...,v \ , \ \forall j = 1,...,n \quad (2)$$

$$\sum_{j=0}^{n} x_{ij}^{k} = y_{i}^{k} \quad \forall k = 1,...,v \ , \ \forall i = 1,...,n \quad (3)$$

$$\sum_{i=0}^{n} y_{i}^{k} \times q_{i} \le Q_{k} \quad \forall k = 1,...,v \quad (4)$$

$$\sum_{k=1}^{v} y_{i}^{k} = 1 \quad \forall i = 1,...,n \quad (5)$$

$$\sum_{k=1}^{v} y_{0}^{k} = v \quad (6)$$

$$t_{i} + W_{i} + s_{i} + t_{ij} = t_{j} \quad \forall i, j = 0,1,..,n, \ i \neq j \quad (7)$$

$$e_{i} \le t_{i} \le l_{i} \quad \forall i = 0,1,2,...,n \quad (8)$$

$$W_{i} = \max\{e_{i} - t_{i}, 0\} \quad \forall i = 0,1,2,...,n \quad (9)$$

Constraints (2)-(3) ensure that each vehicle enters and leaves from any customer if it serves that customer. Constraint (4) verifies that a vehicle capacity is not violated. Constraint (5) verifies that each customer is served one time only by one vehicle. Constraints (6) prescribe that each vehicle journey must start from the depot. Constraints (7)-(9) represent the time constraints; to ensure that time windows is not violated.

3. THE PROPOSED METHOD

This section presents the hybrid algorithm that comprises harmony search and local search algorithm. We first present the harmony search algorithm and its application for the VRPTW. Next, we discuss the hybrid scheme and the three different local search algorithms.

3.1 Harmony Search Algorithm (HSA)

Harmony search algorithm (HSA) is a population-based stochastic search technique that imitates the musical improvisation process in solving a given optimization problem [18]. In the musical improvisation process, a set of musicians play their instrument, step by step while learning from the previous experience, to generate a good harmony. Improvisation process is analogous with generating new solution in the optimization process and this solution will be evaluated by using the objective function [29]. The pitch of each musical instrument in the musician term represents a decision variable in the optimization problems. The role of the improvisation process is to determine the set of the pitch that the musicians will play. The set of pitch are combined together to form a harmony. In optimization, each variable during the optimization process is assigned with a value at a time and combining the entire variable will form a solution, s for the given problem. Similar to the improvisation process, where the harmony is iteratively be improvised to improve its quality; solutions are also iteratively be improved in the optimization process to improve their quality, $f(S)$ [19]. Figure 1 shows the flowchart of the basic HSA, which has five components as follows:

Step 1: Initialization of the HSA parameters. The parameters are:

- Harmony memory size (HMS) that represents the number of solutions to be stored in the harmony memory (HM).
- Harmony memory consideration rate (HMCR) that is used during the improvisation process to generate new solution. Based on the HMCR, the decision variable value is either selected from the HM (with probability HMCR) or randomly initialized (with probability 1-HMCR). HMCR takes a value between zero and one.
- Pitch adjustment rate (PAR) that determine how much we should adjust the selected decision variable of the generated solution to a neighborhood value. PAR takes a value between zero and one.
- The number of improvisations (NI) that represents the number of the iteration or the termination criterion of HSA.

Step 2: Initialization of Harmony memory.
In this step, a set of solutions to a given problem are randomly generated and added to the HM. The number of generated solutions is equal to HMS. In this work, the set of initial solutions for the VRPTW is generated as follows:

a) Create an empty route.
b) Randomly select un-routed customers (that do not violated the VRPTW constraints) and add them to the current route.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

c) If no customer can be added to the current route, a new route is created.
d) The process of creating new route and looping through all customers is repeated until all customers are routed.
e) The quality of the generated solution is calculated using Equation 1 and added to the HM.
f) The process of generating new solution is repeated until the number of generated solutions is equal to the HMS.

Step 3: Improvisation process. In this step, a new harmony (solution) is generated as follows: first create an empty solution *X*. Next, generate a random number *r* between zero and one. If *r* is less than the HMCR, select one route from HM and added it to *X*. Otherwise, create an empty route and a set of customers are randomly assigned to this route as long as the assignment does not violate any constraints. The route is then added to *X*. Next, any route that has been selected from the HM is modified based on the PAR as follows: if the generated random number is less than PAR, randomly select two customers form the current route and swaps their position (if it does not violate VRPTW constrains). The improvisation process will be terminated if the size of new solution *X* (in term of number of routes) is equal to the largest one in HM. If *X* is infeasible in the sense that there is some missed or duplicated customers, duplicated customers are removed, whilst missed customers are either assigned to any feasible route that can accommodate them or create a new route for them.

Step 4: HM updating.
In this step, the quality of the new generated solution *X* is calculated using Equation 1 to replace the worst solution in HM with *X* (if *X* is has better quality than the worst one).

Step 5: Termination process.
This step checks the termination criterion of HSA that is represented by the number of the improvisations (NI). If the maximum number of NI is reached, HSA will stop. Otherwise, start a new improvisation process.

3.2 Hybrid Algorithm

To improve the performance of the harmony search algorithm, we hybridize it with the local search algorithm, as shown in Figure 2. In this hybridization, the local search takes place before the HM updating step. That is, the solution that is generated by the improvisation process is used as an initial solution for the local search algorithm. In this work, we investigate the performance of hybridizing harmony search with three well-known local search algorithms, hill climbing (HC), simulated annealing (SA) and reactive tabu search (RTS). , Then, the resultants are three different hybrid algorithms denoted as HSA-HC, HSA-SA and HSA-RTS. Each local search is used to improve the given initial solution for a predefined number of iterations. The improved solution is then added to the HM, if it better than the worse one in HM. Thus, the local search algorithm is used to intensify the search process in order to improve the harmony search convergence. Integrating a local search algorithm within a population based method has been widely used in the literature to improve the performance of the population based method [6]. The hybridized local search algorithms are:

1- **Simple Hill Climbing HC** (or in minimization, simple descent) starts with an initial solution (*S*). Iteratively generate a neighborhood solution (*S`*) using 2-opt star operator, which randomly select two routes from the current solution and swap the customers located at the end sections of selected routes [3]. If the quality of *S'* is better than *S* replace *S* with *S'* and continue with new iteration. Otherwise, discard *S'* and start new iteration. The search process will be repeated until satisfying stopping criterion, see Figure 3.

2- **Simulated Annealing (SA)**: SA is a local search algorithm proposed by Kirkpatrik [30], SA probabilistically accepts non-improving solution using annealing acceptance criterion in order to escape from the local optima. Given an initial solution (*S*), SA iteratively generates a neighborhood solution (*S`*) using 2-opt star operator [3]. If the quality of *S`* is better than *S*, *S`* *replaces* *S* and new iteration will be started. Otherwise, the worse solution might be accepted with a certain probability *p* calculated as follows: $p = e^{-\Delta f/t}$; where $\Delta f$ is the difference between the quality of *S`* (*f*(*S`*)) and *S* (*f*(*S*)), i.e. $\Delta f = ( f(S`) - f(S))$; and *t* is the current temperature. At each iteration, the temperature is reduced using the geometric scheduled, $g(t)=\beta.t$, where β is the cooling rate ($\beta<1$) [31]. The search process will be repeated until reaching the final temperature, which is usually fixed to a small number. Figure 4 shows a flowchart of SA.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3- **Reactive Tabu Search (RTS)**: Tabu search (TS) is a local search algorithm introduced by Glover [32]. TS uses a tabu list to prevent the search process from revisiting explored region for a certain number of iterations. The use of the tabu list will allow the search to escape from the local optima by visiting new solution that is not in the tabu list [1]. TS work as follows: Given an initial solution (*S*), TS iteratively generates a set of *n* neighbor solutions that is not in the tabu list (in this work, we use 2-opt star operator as in [3]). Then, TS selects the best solution among the *n* generated neighbors. The selected solution or the move's attribute is added into the tabu list and the selected solution is set as an initial solution for the next iteration. TS will update the best solution (if the selected solution has better quality than the best solution). Next, the whole procedures are repeated for a certain number of iterations as shown in Figure 5. The size of the tabu list plays an important role in controlling the diversification and intensification of the search process. Thus, the main parameters that need to be set in advance are the size of tabu list, size of neighborhood and the termination criterion.

In this work, we used a reactive tabu search (RTS) [33]. The search procedure of RTS is similar to the basic TS. The main difference is that the RTS uses the current search state to control the tabu list size. Thus, the size of the tabu list is dynamically changed during the optimization process. RTS monitor the search process and change the size of the tabu list as follows [34]: when the previously visited solutions are revisited again, the size of tabu list should be increased to enhance the diversity of the search. Otherwise, the size of tabu list is decreased to enhance the intensification process [34]. In this work, we implement the RTS for the VRPTW as follows: first, we convert each visited solution into a unique number called *Sr in order to reduce computation time and memory size. Sr* is computed by the Equation 10 [34].

$$Sr = \sum_{p=1}^{v} \sum_{x_j \in R_p} x_j \times |R_p| \qquad (10)$$

However, Equation 10 was proposed for the capacitated vehicle routing problem and we found that different solutions may have the same *Sr* in case of VRPTW. Therefore, in this work, Equation 10 has been modified to overcome this problem. The new version of the equation is (Equation 11):

$$Sr = \sum_{p=1}^{v} \frac{\sum_{j=1}^{|R_p|} x_j \times \sum_{j=1}^{|R_p|} (x_j \times i)}{|R_p|} \qquad (11)$$

Where *Rp* represent the route index, $|Rp|>0$, x_j is the customer in route *Rp*, and *i* is the index of customer x_j in the current solution. Then, the corresponding *Sr* of each visited solution is added to the tabu list as well as the solution age. The solution age represents the number of iteration at which the last visit of the solution was done. By saving *Sr* and the solution age instead of the solution itself, the used storage is reduced and programming complexity is decreased. Then, if an age of any solution in the tabu list reaches the maximum age, the solution is discarded from the tabu list, i.e., decrease the tabu list size.

4 EXPERIMENTAL DESIGN

This section describes the characteristic of the selected Solomon's VRPTW benchmark instances, which are used to evaluate the performance of the hybrid algorithms, basic HSA and the three local search algorithms. The parameter settings for the proposed hybrid algorithm and the local search algorithms are also discussed in this section.

4.1 Solomon's VRPTW Instances

The experiment was performed to assess the performance of the hybrid harmony search and other individual algorithms (HSA, HC, SA and RTS) on eight Solomon's VRPTW instances (see Table 1). These selected instances as shown in Table 1, represents a range of different problem difficulties and size. All algorithms are implemented in Java and executed on a PC running Mac OS with 2.8 GHz Quad-Core Intel Xeon processor and 6 GB RAM.

4.2 Parameter Settings

This section presents the parameter settings for each algorithm that have been used in the experiment. A preliminary test is conducted to determine the suitable parameters values. During the preliminary test, all algorithms (HSA, HC, SA and RTS) are executed 10 runs on eight instances and the best results over 10 runs are reported.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2.1 HSA parameter settings

HSA has four parameters that need to be set in advance. The parameters are: *HMS*, *HMCR*, *PAR* and *NI*. The parameter PAR is fixed as follows (Equation 12):

$$PAR(gn) = PAR_{\max} - \frac{(PAR_{\max} - PAR_{\min})}{NI} \times gn \qquad (12)$$

Where *gn* is the current iteration and *NI* represents the maximum number of iterations. For each iteration, the value of *PAR* decreases dynamically [35]. PAR_{max} is set to 0.9 and PAR_{min} is set to 0.3 as in [29]. Other parameters (i.e. *HMS*, *HMCR* and *NI*) values are set based on the preliminary experiments. The best values for these parameters are: *HMS*=20, *HMCR*=0.7 and *NI*=1000 (based on the best average results, see Table A.1 in Appendix A).

4.2.2 Parameter setting of HC

HC has one parameter only that is the maximum number of iteration (*Max_Iter*). According to the result of testing different *Max_Iter* values, the best average values obtained when *Max_Iter* is fixed to 3000(see Table A.2 in Appendix A).

4.2.3 Parameter settings of SA

SA algorithm has three parameters: the initial temperature (*T_max*), final temprate (*T_min*) and the cooling schedule (β). *T_min* is fixed to 0.5 and β is fixed to 0.99 as suggested by [36]. The *T_max* parameter value was set to 50 depend on preliminary testing (see Table A.3 in Appendix A).

4.2.4 Parameter settings of RTS

RTS has the following parameters: The tabu list size (*TL_size*), the number of neighborhood solutions (*N_neighbors*), maximum number of non-improvement iterations (*MAXI*), the maximum number of iteration allowed for each solution to stay in the tabu list (*Max-age*) and maximum number of iteration (*T_itr*). *TL_size* is dynamically changed depend on the search progress, as explained in Section 3. *T_itr* is fixed to *n**200, where *n* is the number of customers [34]. Based on preliminary testing, RTS perform the best average values when other parameters set to following values: *N_neighbors*=50, *MAXI*=300 and *Max-age*=10 (see Tables A.4-A.6 in Appendix A).

4.2.5 Parameter settings of the hybridized algorithms

The parameter settings of the hybridized algorithm (HSA-HC, HSA-SA and HSA-RTS) are presented in Table 2. The parameters used are based on the best settings on the preliminary tests.

5 ANALYSIS OF RESULTS

The aim of the experimental test in this work is to verify the effectiveness of the integrated local search algorithm with harmony search algorithm. In the conducted experimental test, the computational results of HC, SA, RTS and HAS are compared with the hybrid algorithms (HSA-HC, HSA-SA and HSA-RTS). In this experimental test, the total running time for HC, SA, RTS, HSA and HSA-HC, HSA-SA and HSA-RTS is fixed to be 25 minutes. The stopping criterion for each local search (HC, SA and RTS) inside the hybrid heuristics has been adopted according to the number of non-improved iterations which is 300 iterations.

5.1 The Effect of The Local Search Algorithm

In this section, we compare the results of the local search algorithms (HC, SA, and RTS) with the results obtained by the hybrid algorithms (HSA-HC, HSA-SA and HSA-RTS). For each instance, we report the best (Best), the average (Avr) and the standard deviation (Std) over 31 runs. Tables 3-5 present the results of HC vs. HSA-HC, SA vs. HSA-SA, and RST vs. HSA-RTS. Best results are shown in bold.

From these tables (Tables 3-5), we can see that HSA-HC, HSA-SA and HSA-RTS outperformed HC, SA and RTS over all tested instances with regard to the best, average and standard deviation. To verify whether these results are statistically different, we have conducted a Wilcoxon test with 0.05 critical level. The *p*-value of three different comparisons is presented in Table 6, where the "+" symbol means that the hybrid algorithm is statistically better than the non-hybridize algorithm (*p*-value < 0.05), whilst "-" means otherwise (i.e. *p*-value > 0.05), and "=" means there is no significant differences between the hybrid scheme and the individual algorithm (*p*-value = 0.05). Table 6 proves that, on all tested instances, the hybrid algorithms (HSA-HC, HSA-SA and HSA-RTS) are statistically better than local search algorithms (HC, SA and RTS).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In addition, we illustrates the solution distribution using a box and whisker plot over 31 runs for HC, SA, TS, HSA-HC, HSA-SA, and HSA-RTS (see Appendixes B). The plots show that the solution distribution of the hybrid scheme is much better than individual algorithm on all tested instances.

In summary, the results demonstrate that the use of the local search algorithm within the harmony search does improve the search process. This is mainly because of the local search algorithm good performances in exploiting the neighborhood solution of the current one, whilst harmony search algorithm is good in exploring the search process.

5.2 Comparison of The Proposed Hybrid Algorithms

In this section, we compare standard HSA with the three hybrid schemes (HSA-HC, HSA-SA, and HSA-RTS). Table 7 present, for each instance, the best (Best), average (Avr) and standard deviation (Std) over 31 runs. The second column of this table represents the best known results in the literature. Best results are shown in bold. The tabulated comparison results show that all of the hybridized algorithms obtained the same results on instances C1-01 and C2-01 in terms of the best, average, the standard deviation and the percentage deviation. For the other six instances, the results obtained by the hybrid algorithms can be summarized as follows:

In terms of the best results, the three hybrids schemes (HSA-HC, HSA-SA, and HSA-RTS) outperformed standard HSA on six instances. The standard HSA matched the best results for two instances only (C1-01 and C2-01). All hybrid algorithms obtained the same best result for instance R1-01. For the other five instances, the best results were distributed between HSA-HC and HSA-SA, e.g., HSA-HC obtained the best results for the three instances (R2-01, C2-06 and RC2-01) and HSA-SA obtained the best results in the other two (C1-09 and RC1-01). On average, HSA-HC obtained the best average results on four out of six instances. Whilst, HSA-SA obtained the best average results on two instances. According to the standard deviation results (Std), the same standard deviation was obtained by HSA-HC and HSA-SA on the instance C1-09. For other instances, HSA-HC produced the best standard deviation for the three instances and HSA-RTS on two instances.

Wilcoxon test with 0.05 critical level is performed to proof that the result of each hybrid scheme is statistically different from the standard HSA. The *p*-values of the HSA-HC vs. HSA, HSA-SA vs. HSA and HSA-RTS vs. HAS are presented in Table 8. As can be seen, the results for all hybrid schemes are proven to be statistically better than HSA on all tested instances. The box plot of the solution distribution of the HSA and the three hybrid schemes are presented in Appendix B. The box plot illustrates that the distribution of the three hybrid schemes is better than HSA.

Overall, the presented three hybrid schemes not only obtained better results than the standard HSA but also matched the best known results on two instances (C101 and C2-01) and obtained competitive results compared to the best known results for other tested instances. Therefore, we can conclude that the proposed hybrid algorithms produce good results on the VRPTW problem.

6 CONCLUSION

This work proposed a hybridization of harmony search and local search algorithm for vehicle routing problem with time windows. The harmony search is used to explore the search, whilst local search algorithm is used to further improve the solution generated by the harmony search. Three well-known local search algorithms (hill climbing, simulated annealing and reactive tabu search) were hybridized with the harmony search, which resulting three different hybrid schemes. The proposed hybridized algorithms are tested on the vehicle routing problem with time windows and the results are comparing with the standard harmony search and the local search algorithms. The results demonstrate that all three hybrid schemes produce better results than the compared algorithms. In addition, the hybrid schemes managed to update the best known results in the literature for some instances. The statistical test and the solution distribution show that the integrated local search does help harmony search in producing better solutions. In future work, we intend to investigate the performance of the proposed algorithm on other optimization problems.

REFERENCES:

[1] O. Bräysy and M. Gendreau, "Tabu search heuristics for the vehicle routing problem with time windows," *Top,* vol. 10, pp. 211-237, 2002.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science,* vol. 6, pp. 80-91, 1959.

[3] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part I: Route construction and local search algorithms," *Transportation science,* vol. 39, pp. 104-118, 2005.

[4] Y. J. Gong, J. Zhang, O. Liu, R. Z. Huang, H. S. H. Chung, and Y. H. Shi, "Optimizing the Vehicle Routing Problem With Time Windows: A Discrete Particle Swarm Optimization Approach," *IEEE Transactions on Systems Man and Cybernetics-Part C-Applications Reviews,* vol. 42, p. 254, 2012.

[5] C. B. Cheng and K. P. Wang, "Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm," *Expert systems with applications,* vol. 36, pp. 7758-7763, 2009.

[6] E. G. Talbi, *From design to implementation*: Wiley Online Library, 2009.

[7] S. R. Thangiah, I. H. Osman, and T. Sun, "Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows," *Computer Science Department, Slippery Rock University, Technical Report SRU CpSc-TR-94-27,* vol. 69, 1994.

[8] R. Bent and P. Van Hentenryck, "A two-stage hybrid local search for the vehicle routing problem with time windows," *Transportation Science,* vol. 38, pp. 515-530, 2004.

[9] J. Homberger and H. Gehring, "A two-phase hybrid metaheuristic for the vehicle routing problem with time windows," *European Journal of Operational Research,* vol. 162, pp. 220-238, 2005.

[10] Q. Zhang, T. Zhen, Y. Zhu, W. Zhang, and Z. Ma, "A hybrid intelligent algorithm for the vehicle routing with time windows," *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues,* pp. 47-54, 2008.

[11] B. Yu, Z. Yang, and B. Yao, "A hybrid algorithm for vehicle routing problem with time windows," *Expert Systems with Applications,* vol. 38, pp. 435-441, 2011.

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

[13] Z. J. Czech and P. Czarnas, "Parallel simulated annealing for the vehicle routing problem with time windows," in *Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on*, 2002, pp. 376-383.

[14] B. L. Garcia, J. Y. Potvin, and J. M. Rousseau, "A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints," *Computers & Operations Research,* vol. 21, pp. 1025-1033, 1994.

[15] O. Bräysy and M. Gendreau, "Genetic algorithms for the vehicle routing problem with time windows," *Arpakannus,(1),* pp. 33-38, 2001.

[16] R. A. Russell and W. C. Chiang, "Scatter search for the vehicle routing problem with time windows," *European Journal of Operational Research,* vol. 169, pp. 606-622, 2006.

[17] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, part II: Metaheuristics," *Transportation science,* vol. 39, pp. 119-139, 2005.

[18] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation,* vol. 76, pp. 60-68, 2001.

[19] O. M. Alia and R. Mandava, "The variants of the harmony search algorithm: an overview," *Artificial Intelligence Review,* vol. 36, pp. 49-68, 2011.

[20] M. A. Al-Betar, A. T. Khader, and M. Zaman, "University course timetabling using a hybrid harmony search metaheuristic algorithm," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 42, pp. 664-681, 2012.

[21] Z. Geem, "Harmony search algorithm for solving sudoku," in *Knowledge-Based Intelligent Information and Engineering Systems*, 2007, pp. 371-378.

[22] R. Forsati, M. Mahdavi, M. Kangavari, and B. Safarkhani, "Web page clustering using harmony search optimization," in *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on*, 2008, pp. 001601-001604.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[23] Z. Geem, "Optimal scheduling of multiple dam system using harmony search algorithm," *Computational and Ambient Intelligence,* pp. 316-323, 2007.

[24] W. S. Jang, H. I. Kang, and B. H. Lee, "Hybrid simplex-harmony search method for optimization problems," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 4157-4164.

[25] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems," *Computer methods in applied mechanics and engineering,* vol. 197, pp. 3080-3091, 2008.

[26] Y. C. Lee and A. Y. Zomaya, "Interweaving heterogeneous metaheuristics using harmony search," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, 2009, pp. 1-8.

[27] X. Wang, X. Z. Gao, and S. J. Ovaska, "Fusion of clonal selection algorithm and harmony search method in optimisation of fuzzy classification systems," *International Journal of Bio-Inspired Computation,* vol. 1, pp. 80-88, 2009.

[28] F. W. Takes and W. A. Kosters, "Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem," in *22TH BENELUX CONFERENCE ON ARTIFICIAL INTELLIGENCE (BNAIC)*, 2010.

[29] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 41, pp. 89-106, 2011.

[30] S. Kirkpatrick and M. Vecchi, "Optimization by simmulated annealing," *science,* vol. 220, pp. 671-680, 1983.

[31] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem," *Computers & Operations Research,* 2011.

[32] F. Glover, "Tabu search—part I," *ORSA journal on Computing,* vol. 1, pp. 190-206, 1989.

[33] R. Battiti and G. Tecchiolli, "The reactive tabu search," *ORSA journal on computing,* vol. 6, pp. 126-140, 1994.

[34] N. Wassan and I. Osman, "Tabu search variants for the mix fleet vehicle routing problem," *Journal of the Operational Research Society,* pp. 768-782, 2002.

[35] N. Taherinejad, "Highly reliable harmony search algorithm," in *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on*, 2009, pp. 818-822.

[36] E. K. Burke and G. Kendall, *Search methodologies: introductory tutorials in optimization and decision support techniques*: Springer, 2005.

Journal of Theoretical and Applied Information Technology
10th March 2015. Vol.73 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

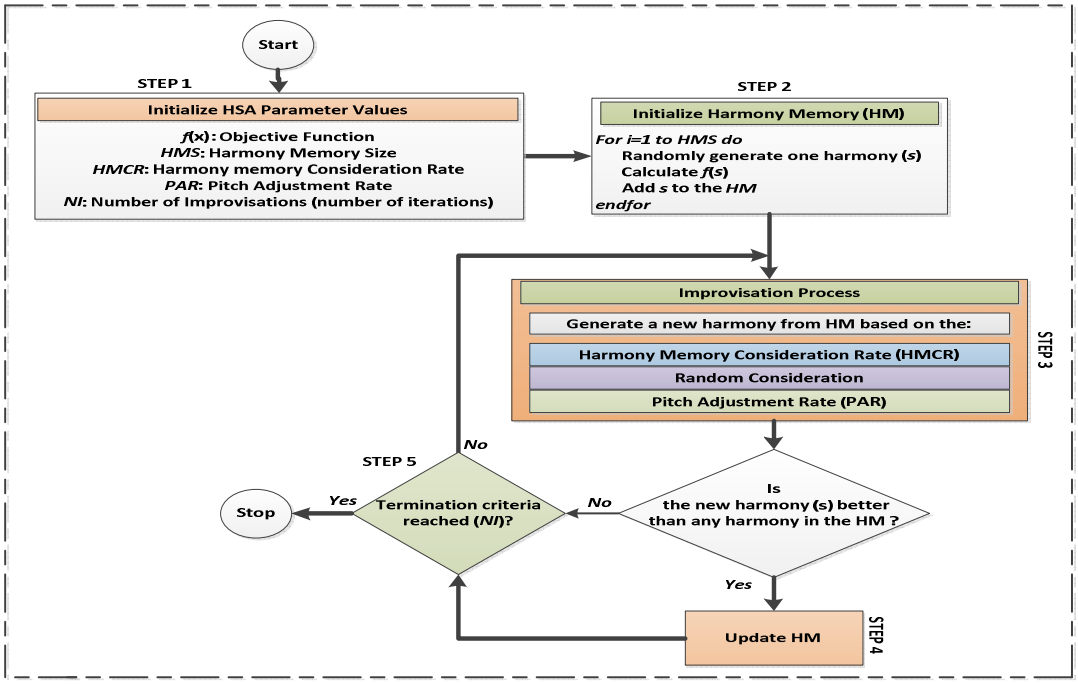

Figure 1 Hsa Flowchart

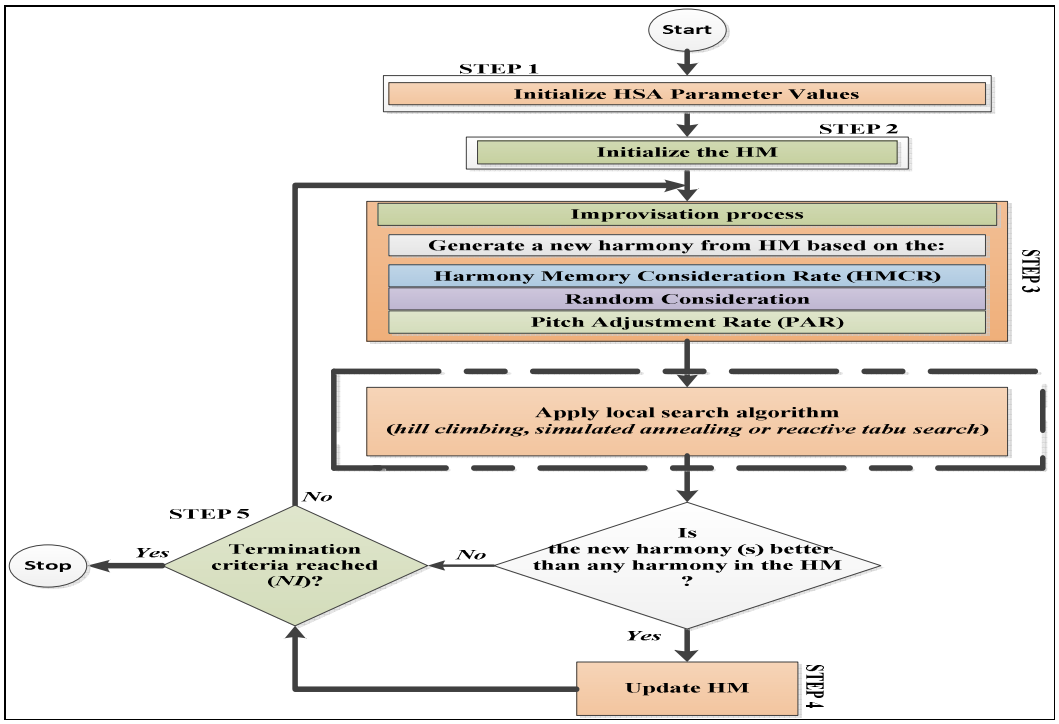

Figure 2 The Hybrid Harmony Search

Journal of Theoretical and Applied Information Technology
10th March 2015. Vol.73 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

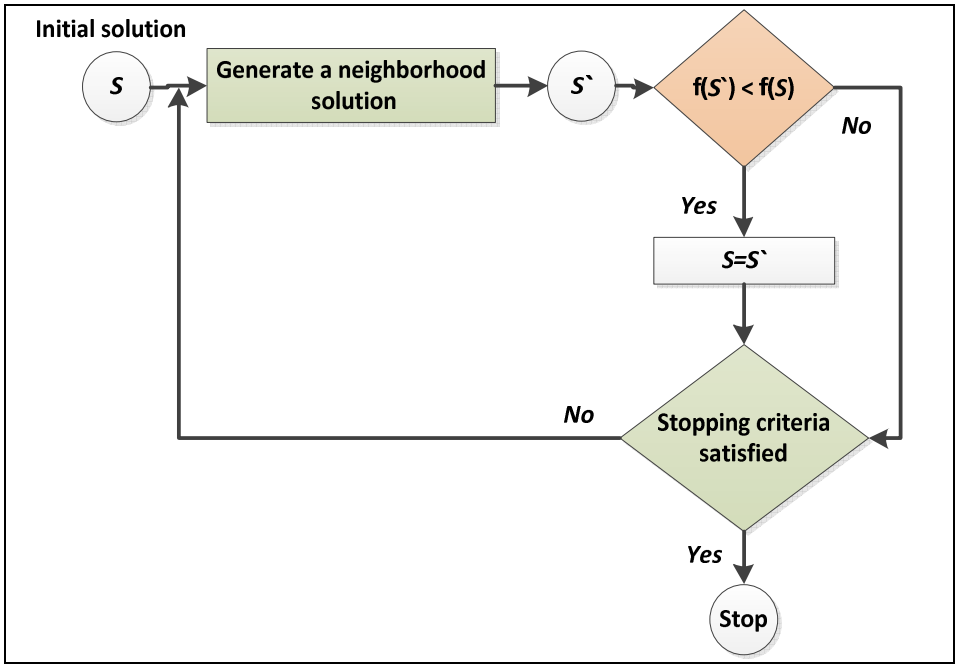

Figure 3 Hill Climbing Algorithm

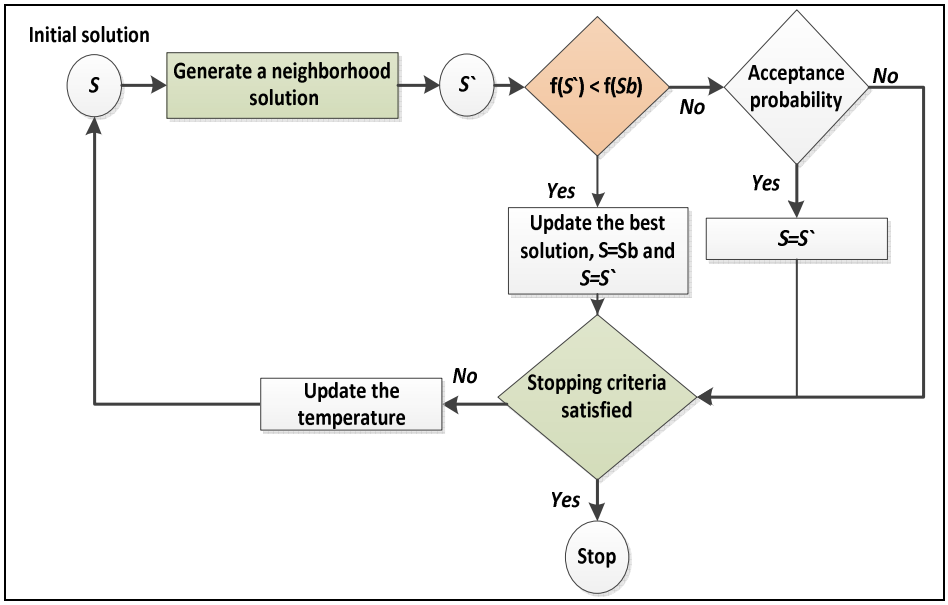

Figure 4 Simulated Annealing Algorithm

Journal of Theoretical and Applied Information Technology
10th March 2015. Vol.73 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

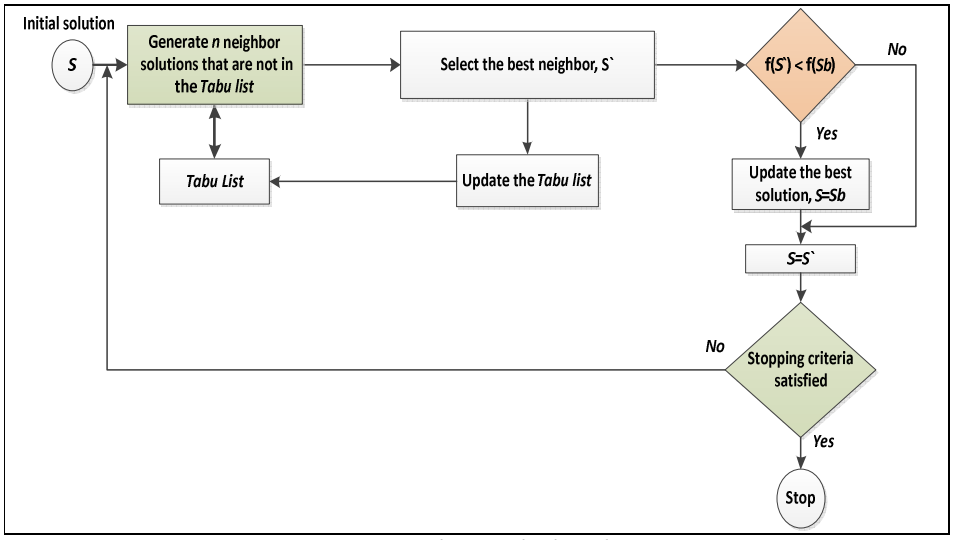

Figure 5 Tabu Search Algorithm

Table 1 The Characteristics Of Selected Solomon's VRPTW Instances

Instances	No. of Customers	No. of Vehicles	Capacity of Vehicle	Distribution of Customers	Width of Time Window
R1-01	100	25	200	Random	Small Time Windows
R2-01	100	25	1000	Random	Large Time Windows
C1-01	100	25	200	Cluster	Small Time Windows
C1-09	100	25	200	Cluster	Small Time Windows
C2-01	100	25	700	Cluster	Large Time Windows
C2-06	100	25	700	Cluster	Large Time Windows
RC1-01	100	25	200	Random /Cluster	Small Time Windows
RC2-01	100	25	1000	Random /Cluster	Large Time Windows

Table 2 The Parameters Setting Of Hybridized Algorithm

Parameter	Algorithm	Value
HMS	HSA	20
HMCR	HSA	0.7
NI	HSA	1000
PAR_max	HSA	0.9
PAR-min	HSA	0.3
Max_Iter	HC	3000
T_max	SA	50
T_min	SA	0.5
β	SA	0.99
T_itr(RTS)	RTS	*n**200
MAXI	RTS	300
Max_age	RTS	10
N_neighbors	RTS	50

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3 Comparison Between HC And HSA-HC

Dataset	HC			HSA-HC		
	Best	Avr	Std	Best	Avr	Std
R1-01	1707.34	1782.48	38.01	**1642.88**	**1642.89**	**0.05**
R2-01	1293.50	1346.26	32.26	**1203.61**	**1225.69**	**9.87**
C1-01	**828.94**	949.90	64.15	**828.94**	**828.94**	**0.00**
C1-09	1013.16	1140.34	65.73	**831.79**	**839.51**	**4.08**
C2-01	632.05	735.26	52.42	**591.56**	**591.56**	**0.00**
C2-06	745.20	803.82	41.18	**644.32**	**675.80**	**12.34**
RC1-01	1808.73	1894.29	39.86	**1632.20**	**1650.62**	**7.61**
RC2-01	1433.29	1534.54	46.79	**1326.45**	**1357.72**	**14.54**

Table 4 Comparison Between Sa And Hsa-Sa

Dataset	SA			HSA-SA		
	Best	Avr	Std	Best	Avr	Std
R1-01	1698.15	1776.16	47.53	**1642.88**	**1642.88**	**0.00**
R2-01	1297.18	1352.27	26.77	**1206.57**	**1233.43**	**10.40**
C1-01	**828.94**	966.00	61.27	**828.94**	**828.94**	**0.00**
C1-09	1045.99	1132.90	51.91	**831.30**	**840.47**	**4.08**
C2-01	631.12	740.31	50.89	**591.56**	**591.56**	**0.00**
C2-06	721.34	795.92	43.32	**650.44**	**674.52**	**15.86**
RC1-01	1819.22	1909.01	60.30	**1631.17**	**1650.68**	**8.80**
RC2-01	1424.36	1532.97	55.22	**1332.09**	**1359.49**	**13.88**

Table 5 Comparison Between RST And HSA-RST

Dataset	RTS			HSA-RTS		
	Best	Avr	Std	Best	Avr	Std
R1-01	1729.07	1785.51	39.51	**1642.88**	**1645.97**	**1.92**
R2-01	1276.12	1355.83	40.10	**1258.61**	**1281.18**	**13.64**
C1-01	**828.94**	966.27	61.66	**828.94**	**828.94**	**0.00**
C1-09	969.53	1119.15	68.60	**866.40**	**894.73**	**10.71**
C2-01	667.46	736.47	32.64	**591.56**	**591.56**	**0.00**
C2-06	738.29	801.37	39.47	**699.22**	**731.24**	**19.41**
RC1-01	1803.87	1880.34	48.65	**1660.36**	**1679.53**	**8.90**
RC2-01	1455.60	1530.49	41.55	**1366.48**	**1426.76**	**24.72**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6 The P-Value Of The Wilcoxon Test

Instances	HSA-HC *vs* HC	HSA-SA *vs* SA	HSA-RTS *vs* RTS
	P-value	*P-value*	*P-value*
R1-01	+	+	+
R2-01	+	+	+
C1-01	+	+	+
C1-09	+	+	+
C2-01	+	+	+
C2-06	+	+	+
RC1-01	+	+	+
RC2-01	+	+	+

Table 7 The results of HAS, HSA-HC, HSA-SA and HSA-RTS

Instances	Best known	HSA			HSA-HC			HSA-SA			HSA-RTS		
		Best	Avr	Std	Best	Avr	Std	Best	Avr	Std	Best	Avr	Std
R1-01	1642.87	1690.39	1741.15	21.65	**1642.88**	1642.89	0.05	**1642.88**	**1642.88**	**0.00**	**1642.88**	1645.97	1.92
R2-01	1148.48	1419.44	1521.87	37.16	**1203.61**	**1225.69**	**9.87**	1206.57	1233.43	10.40	1258.61	1281.18	13.64
C1-01	828.94	**828.94**	990.88	103.84	**828.94**	**828.94**	**0.00**	**828.94**	**828.94**	**0.00**	**828.94**	**828.94**	**0.00**
C1-09	828.94	1099.79	1225.69	64.80	831.79	**839.51**	**4.08**	**831.30**	840.47	**4.08**	866.40	894.73	10.71
C2-01	591.56	**591.56**	724.42	99.52	**591.56**	**591.56**	**0.00**	**591.56**	**591.56**	**0.00**	**591.56**	**591.56**	**0.00**
C2-06	588.49	817.02	888.88	43.71	**644.32**	675.80	**12.34**	650.44	**674.52**	15.86	699.22	731.24	19.41
RC1-01	1623.58	1713.62	1800.66	37.90	1632.20	**1650.62**	**7.61**	**1631.17**	1650.68	8.80	1660.36	1679.53	8.90
RC2-01	1274.54	1689.17	1772.53	38.89	**1326.45**	**1357.72**	14.54	1332.09	1359.49	**13.88**	1366.48	1426.76	24.72

Table 8 The p-value of the Wilcoxon test HSA-HC, HSA-SA, HSA-RTS against standard HSA

Instances	HSA-HC vs HSA	HSA-SA vs HSA	HSA-RTS vs HSA	HSA-HC vs HSA-SA	HSA-HC vs HSA-RTS	HSA-SA vs HSA-RTS
	P-value	P-value	P-value	P-value	P-value	P-value
R101	+	+	+	-	+	+
R201	+	+	+	+	+	+
C101	+	+	+	=	=	=
C109	+	+	+	-	+	+
C201	+	+	+	=	=	=
C206	+	+	+	-	+	+
RC101	+	+	+	-	+	+
RC201	+	+	+	+	-	+

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Appendix A

Preliminary Test

Table A.1 The Results Of Different HSA Parameter Values

Instances	HMS			Instances	HMCR			Instances	NI		
	10	20	30		0.6	0.7	0.8		500	1000	2000
R101	2103.7	**2074.47**	2088.89	R101	2095.49	2074.47	**2056.56**	R101	2087.99	**2074.47**	2086.81
R201	1548.25	**1496.51**	1544.35	R201	1542.11	**1496.51**	1512.29	R201	1524.8	**1496.51**	1528.99
C101	1720.52	**1661.05**	1703.99	C101	1727.96	**1661.05**	1706.84	C101	1698.68	**1661.05**	1726.21
C201	683.23	**591.56**	824.89	C201	742.89	**591.56**	666.38	C201	809.85	**591.56**	603.88
RC101	2287.16	2292.89	**2244.2**	RC101	**2241.98**	2292.89	2267.29	RC101	**2201.64**	2292.89	2254.9
RC201	1777.79	**1744.75**	1765.92	RC201	1846.68	**1744.75**	1792.41	RC201	1823.57	**1744.75**	1807.54
Average	1686.78	**1643.54**	1695.37	Average	1699.52	**1643.54**	1666.96	Average	1691.09	**1643.54**	1668.06

Table A.2 The Results Of Different Max_Itr Value

HC Max_Itr Values						
Instances	1000	2000	3000	4000	5000	6000
R1-01	1721.49	**1705.62**	1743.39	1719.64	1749.84	1727.87
R2-01	1280.98	1310.63	1300.09	1316.54	**1251.01**	1310.43
C1-01	**828.94**	866.76	**828.94**	889.97	892.25	892.25
C2-01	656.07	683.54	**626.05**	633.95	631.12	654.08
RC1-01	1829.18	1844.71	**1768.45**	1775.21	1836.64	1796.94
RC2-01	1461.27	1457.03	1460.76	1456.34	1444.45	**1432.26**
Average	1296.32	1311.38	**1287.95**	1298.61	1300.89	1302.31

Table A.3 Best Results Of Different T_Max Values

Initial Temperature (T_max)			
Instances	30	50	100
R101	1803.35	**1760.53**	1768.09
R2-01	1304.5	**1291.11**	1312.14
C1-01	**982.5**	993.34	983.97
C2-01	705.03	**653.26**	666.35
RC1-01	1925.08	1919.09	1889.621
RC2-01	1445.01	**1432.37**	1464.68
Average	1360.912	**1341.617**	1347.475

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table A.4 The Results Of The N_Neighbors Parameter

N_neighbors values						
Instances	300	200	100	50	25	10
R101	1720.89	1769.09	1752.7	**1709.65**	1726.63	1751.74
R2-01	1309.79	1301.85	1308.41	**1271.51**	1279.68	1307.3
C1-01	866	895.42	**828.94**	**828.94**	**828.94**	892.25
C2-01	684.08	**633.89**	677.65	679.18	662.58	694.34
RC1-01	1772.72	1813.1	1769.77	**1727.38**	1777.08	1790.31
RC2-01	**1439.21**	1492.5	1447.14	1484.42	1461.85	1448.54
Average	1214.36	1227.35	1206.38	**1198.29**	1202.03	1226.59

Table A.5 The Results Of Testing Different MAXI Values

MAXI values							
Instances	400	300	200	100	50	25	10
R101	1743.8	1726.63	**1724.55**	1755.97	1725.58	1735.43	1735.44
R2-01	1310.47	**1279.68**	1314.07	1321.07	1290.95	1293.91	1304.41
C1-01	866	**828.94**	889.45	906.85	899.49	889.97	889.45
C2-01	690.4	**662.58**	680.38	676.23	695.87	677.16	**662.58**
RC1-01	1791.58	1777.08	1791.74	1835.82	1833.21	**1759.08**	1786.16
RC2-01	1477.75	1461.85	1494.2	1439.44	1458.08	**1425.35**	1440.87
Average	1227.24	**1202.03**	1233.97	1235.88	1235.52	1209.09	1216.69

Table A.6 The Results Of Testing Different Max_Age Values

Max_age values						
Instances	300	200	100	50	25	10
R101	1762.87	1739.1	**1694.13**	1729.72	1710.38	1726.63
R2-01	1297.48	1317.43	**1285.34**	1304.08	1292.73	1279.68
C1-01	889.45	895.86	896.71	933.2	866.87	**828.94**
C2-01	666.55	682.9	699.61	728.97	708.58	**662.58**
RC1-01	1827.15	1809.27	1779.97	1834.76	1815.28	**1777.08**
RC2-01	1464.97	**1459.83**	1499.37	1461.52	1469.85	1461.85
Average	1229.12	1233.06	1232.2	1252.51	1230.66	**1202.03**