วิศวกรรมสาร มข. ปีที่ 38 ฉบับที่3 (313 -323) กรกฎาคม – กันยายน 2554 บทความวิจัย

KKU Engineering Journal Vol.38 No3 (313 -323) July – September 2011

Max-Min Ant System (MMAS) for Vehicle Routing Problem with Time Windows

Suphan Sodsoon*1) and Preecha Changyom2)

Abstract

This paper introduces a modified Max-Min Ant System (MMAS) algorithm to solve the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW can be described as the problem of designing least cost routes from one depot to a set of geographically scattered points. The routes must be designed in such a way that each point is visited only once by exactly one vehicle within a given time interval, all routes start and end at the depot, and the total demands of all points on one particular route must not exceed the capacity of the vehicle. Many meta-heuristic approaches like Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS), a hybrid Ant System algorithm for VRP (HAS-VRP), a multiple Ant Colony System to vehicle routing problem with time windows (MACS-VRPTW) and a hybrid Ant System algorithm and Simulated Annealing (IACS-SA). In this research, we proposed a MMAS-VRPTW algorithm with local search approaches. Experiments on various aspects of Solomon's 56 benchmark problems are other meta-heuristic and show that our results are competitive.

Keywords: vehicle routing problem with time windows meta-heuristic max-min ant system

1)Lecturer, Industrial Technology, Rajamangala University of Technology Isan, Kalasin Campus, Kalasin 46000, E-mail: suphan_sdosoon@hotmail.com

2) Assistant Professor, Industrial Engineering, Rajamangala University of Technology Lanna, Chaingmai 50300 E-mail: changchana@hotmail.com

* Corresponding Author

1. Introduction

The vehicle routing problem with time windows (VRPTW) is an extension of the vehicle routing problem with earliest, latest and service times for customers. The VRPTW routes a set of vehicles to service customers having earliest, latest service times. The objective of the problem is to minimize the number of vehicles and the distance travelled to service the customers. The constraints of the problem are to service all the customers after the earliest release time and before the latest service time of each customer without exceeding the route time of the vehicle and overloading the vehicle. The route time of the vehicle is the sum total of the waiting time, the service time and distance travelled by the vehicle. A vehicle that reaches a customer before the earliest release time incurs waiting time. If a vehicle services a customer after the latest delivery time the vehicle is considered to be tardy. The service time is the time taken by a vehicle to service a customer. A vehicle is said to be overloaded if the sum total of the customer demands exceed the total capacity of the vehicle. The quality of the solution is measured in terms of the minimization of the number of vehicles followed by the minimization of the total distance travelled respectively in that order. That is, a solution for the VRPTW with a lower total number of vehicles and greater total distance travelled is preferred over a solution that requires greater number of vehicles and smaller total distance travelled.

2. Reviewed Literature

The VRPTW has been the subject of intensive research efforts for both heuristic and exact optimization approaches. Early surveys of solution techniques for the VRPTW can be found in Desrochers et al (1992), focus on exact techniques. Further details on these exact methods can be found in Lin. (1965). Because of the high complexity level of the VRPTW and its wide applicability to real-life situations, solution techniques capable of producing high-quality solutions in limited time, i.e., heuristics are of prime importance. Over the last few years, many authors have proposed new heuristic approaches, mostly meta-heuristics, for tackling the VRPTW. Moreover, the VRPTW has been proved to be NP-hard and exact algorithms cannot find the optimal solution for large VRPTW within reasonable computational times. Thus many heuristic approaches have been proposed in the literature. Many meta-heuristic approaches like Simulated Annealing (Czech and Czarnas (2002); Li and Lim (1965)), Genetic Algorithms (Potvin and Bengio (1996); Berger et al.(2001); Chen et al.(2004); Tan et al.(2001); Ting and Huang (2005)), Tabu Search (Chiang and Russell (1997); Taillard et al.(1997)), Gambardella et al.(1999) defined a hybrid Ant System algorithm for VRP (HAS-VRP), which was inspired by ACS. Results obtained by HAS-VRP were competitive with those of the best-known algorithms and new upper bounds have been found for well-known problem instances. Furthermore, Gambardella et al (1999) proposed a multiple Ant Colony System to vehicle routing problem with time windows (MACS-VRPTW) and improved some of the best-known solutions in the literature.

3. Problem Formulations

The VRPTW is defined on a graph (N, A). The node set N consists of the set of customers, denoted by C, and the nodes 0 and $n+1$, which represent the depot. The number of customers $|C|$ will be denoted n and the customers will be denoted by $1,2,\ldots,n$.

The arc set A corresponds to possible connections between the nodes. No arc terminates at node 0 and no arc originates at node n+1. All routes start at 0 and end at n+1. A cost c_{ij}

$$\text{minimize} \quad \sum_{k\in V}\sum_{(i,j)\in A} c_{ij} X_{ij}^{k} \tag{1}$$

subject to :

$$\sum_{k\in V}\sum_{j\in N} X_{ij}^{k} = 1, \qquad \forall i \in C \tag{2}$$

$$\sum_{i\in C} d_i \sum_{j\in N} X_{ij}^{k} \le q, \qquad \forall k \in V \tag{3}$$

$$\sum_{j\in N} X_{0j}^{k} = 1, \qquad \forall k \in V \tag{4}$$

$$\sum_{i\in N} X_{ih}^{k} - \sum_{j\in N} X_{hj}^{k} = 0, \qquad \forall h \in C,\ \forall k \in V \tag{5}$$

$$\sum_{i\in N} X_{i,n+1}^{k} = 1, \qquad \forall k \in V \tag{6}$$

$$X_{ij}^{k}(S_i^k + t_{ij} - S_j^k) \le 0, \qquad \forall (i,j) \in A, \forall k \in V \tag{7}$$

$$a_i \le S_i^k \le b_i, \qquad \forall i \in N, \forall k \in V \tag{8}$$

$$X_{ij}^{k} \in \{0,1\}, \qquad \forall (i,j) \in A, \forall k \in V \tag{9}$$

and travel time t_{ij} are associated with each arc (i, j)$\in$ A of the network. The travel time t_{ij} includes a service time at customer i. The set of (identical) vehicles is denoted by V. Each vehicle has a given capacity q and each customer a demand d_i, $i \in C$. At each customer, the start of the service must be within a given time interval, called a time window, [a_i, b_i], $i \in C$. Vehicles must also leave the depot within the time window [a_0, b_0] and return during the time window [a_{n+1}, b_{n+1}]. A vehicle is permitted to arrive before the opening of the time window, and wait at no cost until service becomes possible, but it is not permitted to arrive after the deadline. Since waiting time is permitted at no cost, we may assume without loss of generality that $a_0 = b_0 = 0$; that is, all routes start at time 0. The model contains two types of decision variables. The decision variable X_{ij}^{k} (defined $\forall (i, j) \in A, \forall k \in V$) is equal to 1 if vehicle k drives from node i to node j, and 0 otherwise. The decision variable S_i^k (defined $\forall i \in N, \forall k \in V$) denotes the time vehicle k, $k \in V$, starts service at customer i, $i \in C$. If vehicle k does not service customer i, S_i^k has no meaning. We may assume that $S_0^k = 0, \forall k$, and S_{n+1}^k denotes the arrival time of vehicle k at the depot. The objective is to design a set of minimal cost routes, one for each vehicle, such that all customers are serviced exactly once. Hence, split deliveries are not allowed. The routes must be feasible with respect to the capacity of the vehicles and the time windows of the customers serviced. The VRPTW can be stated mathematically as: The objective function (1) states that costs should be minimized. Constraint set (2) states that each customer must be assigned to exactly one vehicle, and constraint set (3) states that no vehicle can service more customers than its capacity permits. Constraint set (4), (5) and (6) are the flow constraints requiring that each vehicle k leaves node 0 once, leaves node h, $h \in C$, if and only if it enters that node, and returns to node n+1. Note that constraint set (6) is redundant, but is maintained in the model to underline the network structure. The arc (0, n+1) is included in the network to allow empty tours. More precisely, we permit an unrestricted number of vehicles, but a cost c_v is put on each vehicle used. This is done by setting $c_{0,n+1} = -c_v$. The value of c_v is sufficiently large to primarily minimize the number of vehicles and secondarily minimize travel costs. Nonlinear (easily linearized, see for example Desrosiers et al., 1995) constraint set (7) states that vehicle k cannot arrive at j before $S_i^k + t_{ij}$ if it travels from i to j. Constraint set (8) ensures that all time windows are respected and (9) is the set of integrality constraints.

4. MMAS to Solve the VRP

MMAS algorithm derived from Ant System that it developed by Stützle and Hoos (1999). MMAS achieves a strong exploitation of the search history by allowing only the best solutions to add pheromone during the pheromone trial update. Also, the use of a rather simple mechanism for limiting the strengths of the pheromone trails effectively avoids premature convergence of the search. MMAS-VRPTW is described as follows:

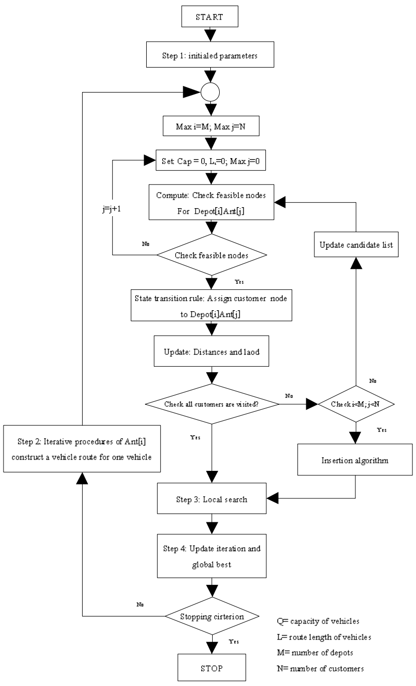

Fig. 2 *MMAS-VRPTW algorithms*

Step 1, the procedure finds a feasible solution by an algorithm based on nearest neighbor. This algorithm has been chosen because of its low complexity, even if it does not guarantee all nodes to be inserted in the tour: it may happen, owing to particular time windows, that some nodes are not included. Since the aim of this step is just providing a rough estimation of the cost of the tour, needed to fix the initial level of the pheromone on edges, this problem has been faced by computing the average length of an edge used in the partial solution and adding to the total cost the length of as many edges as are not yet visited. Let ω^{gb} be the current best solution (globally best).

Step 2, requires that a colony of ants is activated to find the shortest route. The first step is to determine the initialized a number of vehicles (n_v), capable to cope with all demand. (n_v is simply determined by the total demand divided by vehicle capacity). In this research, we used the amount of ant colonies equal to the number of vehicles $(n_v + 1)$ to construct routes as equation (10).

$$n_v = \left\lfloor \frac{T_d}{L_v} \right\rceil \qquad (10)$$

Where, n_v is number of vehicle, T_d is total demand of customer and L_v is maximum load of vehicle.

The idea of the algorithm is to construct only feasible solutions using as multi colonies. In every generation, each ant k (a set of k^{th}) constructs one feasible solution, starting at the depot and successively choosing a next node or customer j, from the set of feasible nodes. Ant ants works can be analyzing each node with respect to the constraints imposed by the model, each ant builds list of feasible movements and chooses the one indicated by the probabilistic rule. As explained before considering the first step, it is not guaranteed that all the nodes can be easily inserted in the tour, exploiting collective learning to reach this results, following Gambardella et al. (1996), the best tour is, firstly, the one that reaches the highest number of nodes. Global pheromone update is made using the best solution found in this way. We extension of the Solomon algorithm (1987) for ant colony construct routes in two frameworks, sequential and parallel constructions and theses algorithms are illustrated in Fig. 3-4, respectively. The parallel construction terminates, when there is no more demand left. For an ant construction routes, after we known a number of multi-colonies and set it positioned on each vehicle for an ant colonies construct vehicles routes by alternating motion of each ant from each depot.

An ant selects the next customer to be served, compatible with capacity constraints. Each ant is put at a depot and each ant will choose next nodes to move from the present node i to the next node j according to the state transition rule. In the parallel route construction shown in Fig. 4, the first route is constructed for every vehicle, then, the second route, the third, and so on. However, not every tour may have the same number of routes, since there may be no more feasible points to be included during the construction of any route for later vehicles or routes. In such case, the next loop is repeated for a new route to be formed for every vehicle if possible. In Fig. 3, where, $currenttime$ is the current time, $\delta_j = \max\{current + t_{ij}, e_j\} - current$ is the width of the time interval elapsing before the beginning of the service to customer j. MMAS-VRPTW algorithm can be produce infeasible solutions in which some customers are not visited. The solution with highest number of visited customers is stored in MMAS-VRPTW algorithm. If the solution is incomplete at the end of the constructive phase, all non visited customers sorted in decreasing deliver quantities are inserted to the best feasible location.

Step 3, Route improvement

The route improvement procedure starts from an initial solution obtained from the route construction phase and attempts to and a better neighboring solution in terms of the number of vehicles and total time spent, while maintaining solution feasibility.

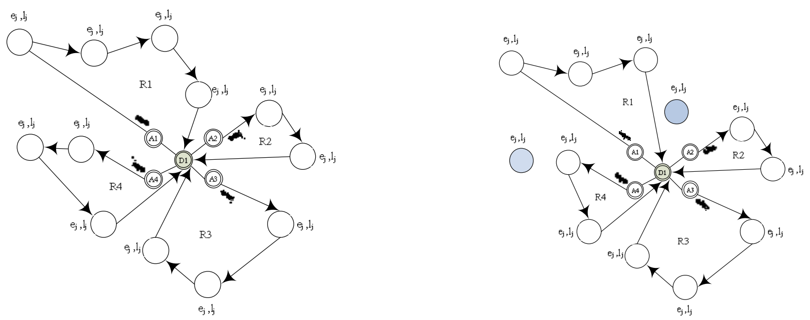

Fig. 3 *Feasible and infeasible solutions for a with 4 ants or 4 vehicles*

In this paper we employed three local search procedures, namely Move-Exchanges, Or-opt and 2-opt algorithms which have been popular among exchange techniques proposed for solving VRP. Especially, these three methods, after exchanges, still maintain the movement orientation. After an ant has constructed its solution, we apply a local search algorithm to improve the solution quality. In particular, we apply Move-Exchanges, Or-opt and 2-opt algorithm were combined in the improvement procedure to take advantage of their own strength. The overall improvement procedure of local search in MMAS-VRPTW is as follows in Fig. 4:

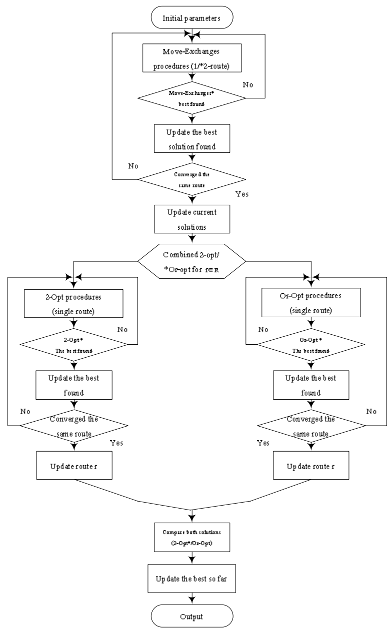

Fig. 4 *Route improvement procedures*

Type I. Move-Exchange modification

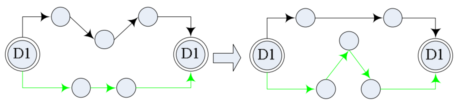

Fig. 5 *Move-Exchange algorithm*

Type II. 2-opt algorithm

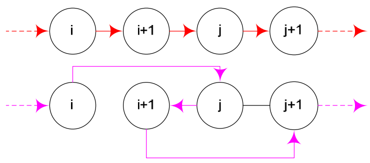

Fig. 6 *2-Opt algorithm*

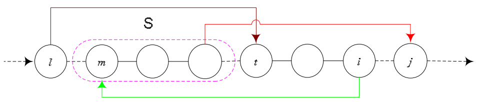

Fig. 7 *Or-opt algorithm*

The Move-Exchanges operator aims at improving the solution by exchanging a customer i with a customer j by tire to eject a customer i from its current position and insert it at another position.

The edge-exchange neighborhoods for a single route are set of route that can be obtained from an initial route by replacing a set of k of its edges by another set of k edges. Such replacements are called k-exchanges. 2-exchanges or 2-opt is illustrated in Figure 6. The edges $(i,i+1)$ and $(j, j+1)$ are replaced by edges (i, j) and $(i+1, j+1)$. Thus the direction of customers between $i+1$ and j is reverse. It tries to improve the route by replacing two of its edges by two other edges and iterations until no further improvement is possible.

Step 4, Updating pheromone trails

The MMAS-VRPTW to update pheromone trails includes iteration-best and global-best solutions to avoid search stagnation. The allowed range of the pheromone trails strength is limited to the interval $[\tau_{\max}, \tau_{\min}]$, that τ_{ij} is $\tau_{\min} \le \tau_{ij} \le \tau_{\max}$. The pheromone trails are initialized to the upper trail limits. After all ants have constructed solutions, the pheromone trails are updated according to $\tau_{ij}(t+1) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}$. Where ρ, a parameter is called evaporation coefficient, $0 < \rho < 1$ and $\Delta\tau_{ij}^{best} = 1/C^{best}$ where t is scheduled for the frequency and C^{best} is the best so far tour. The ant which is allowed to add pheromone trails may construct iteration-best tour and global-best tour. All edges (i, j) belonging to the so far best solution (objective value) are considered to increase the intensity of pheromone trails by an amount $\Delta\tau_{ij}^{best}$. If edges (i, j) does not belong to the so far best solution, the intensity of pheromone will be reduced.

5. Numerical Analysis

In this section we will present numerical results for our new approach and compare them with results from previous literature as well as different meta-heuristics. We conduct computational experiments on Solomon's 56 benchmark problems. These problems were generated in six classes: R1, R2, C1, C2, RC1 and RC2. All problems have 100 customers, a central depot, capacity constraints and time window constraints. The customers are randomly distributed in R1 and R2 problems, while in C1 and C2 problems they are clustered. In RC1 and RC2 problems, customers are mixed with both clustered and randomly distributed. The C1, R1 and RC1 problems have short scheduling horizon, while C2, R2 and RC2 have longer scheduling horizon. We summarize information of the Solomon benchmark problems in Table 1. Columns 2-7 show the number of problems in each type, the vehicle capacity, schedule horizon, service times of each node, the node distribution and the elasticity of time window.

Table 1 *The information of Solomon's VRPTW benchmark problems*

Type	Number of problems	Vehicle capacity	Schedule horizon	Service time	Node distribution	Time windows
R1	12	200	230	10	randomly	narrow
R2	11	1000	1000	10	randomly	large
C1	9	200	1236	90	clustered	narrow
C2	8	700	3390	90	clustered	large
RC1	8	200	240	10	Mixed	narrow
RC2	8	1000	960	10	Mixed	large

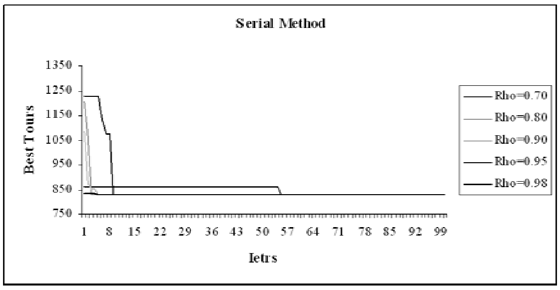

Fig. 8 *Serial Method with C101*

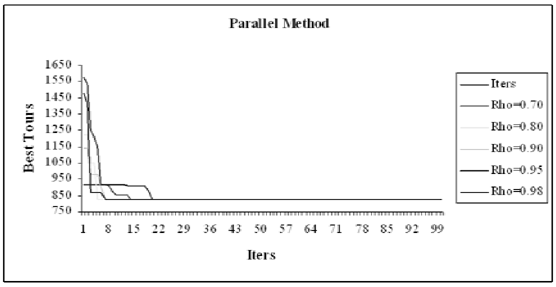

Fig. 9 *Parallel Method with C101*

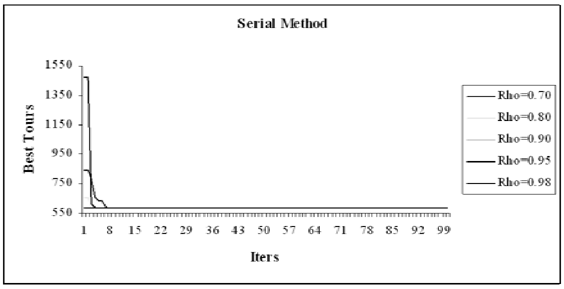

Fig. 10 *Serial Method with C201*

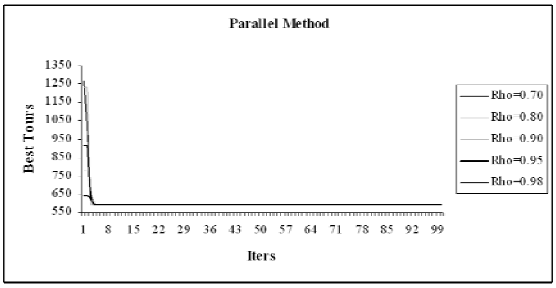

Fig. 11 *Parallel Method with C201*

6. Computational results

The MMAS-VRPTW is coded in Visual C++ version 5.0 and executed on a Pentium IV, 256 MB of RAM, 3.07 GHz processor. The MMAS parameters used for VRPTW instances are α =1, β =3, γ =0.90 and I^{max} = N and solve 5 times for each problem. In this section we experimentally study the effectiveness of the influence of difference values of the parameters. To determine the appropriate values of parameters β, α, ρ and I^{Max} ,which determines the convergence speed of MMAS towards good solution. Preliminary tests have been performed for difference values of β, α, ρ and I^{Max}. The values tested were $\alpha = 1, \beta = 2$ $\alpha = 0.7 - 0.98$ and I^{Max} =500 and all the combinations were checked. The analysis indicated that satisfactory can be presented curves for the trade-off between the best solutions is found versus serial method and parallel method on routes building for C101 and C201 instances by using difference setting of ρ is varies between 0.7 and 0.98, shown in Fig. 8-11. Concerning the solution construction, there are two different ways implementing which are serial and parallel solution constructions. From the results, serial and parallel give different results. The way to construct solution has significant effect on solution quality. In the cases, serial constructions provide the minimum total distances when compared with other implementation.

Table 2 *Percentage deviation and number of new best solutions*

Pro. Type	Best Known (Distance)	MMAS-VRPTW(Distance)	%RPD
C1	828.310	838.120	1.770
C2	589.860	519.130	0.214
R1	1209.890	1259.190	4.147
R2	951.910	980.983	3.170
RC1	1384.160	1436.584	4.090
RC2	1119.350	1141.626	2.820
Avg.	1013.913	1029.272	2.702

Therefore serial approach was most appropriated ant releasing method for all set of problem instances. Percentage deviation and number of new best solutions are presented in Table 2. The algorithms considered are: the Tabu Search (TS-P) of Potvin et al. (1996), the Tabu Search (TS-T) of Taillard et al. (1997), the Multiple Ant Colony System (MACS) of Gambardella et al. (1999), the Genetic Algorithm (GA) of Berger et al. (2001), the Hybrid Genetic Algorithm (HGA) of Chen et al. (2001), the hybrid SA and TS (SATS) of Tan et al. (2001) and the tabu-embedded SA (TESA) of Li and Lim (2003). It is also shown in Table 3-4. IACS-SA yields best results for R1, R2, C2, RC1 and produces the lowest cumulated total traveled distance among eight algorithms. Taillard's TS and Gambardella's MACS obtained the best results for C1 and C2. Comparison of MMAS-VRPTW with RMACS-VRPTW, IACS-SA and Best known solutions.

Table 4 *Comparison of MMAS-VRPTW with other meta-heuristics*

Problem Type	C1	C2	R1	R2	RC1	RC2
	10.00	3.00	12.60 a	3.10	12.60	3.40
	861.00	602.50	1294.70 b	1185.90	1465.00	1476.10
TS-P1	435.00	431.00	639c	722.00	586.00	662.00
	10.00	3.00	12.17	2.82	11.50	3.38
	828.38	589.86	1209.35	980.27	1389.22	1117.44
TS-T2	14630.00	16375.00	13774.00	20232.00	11264.00	11596.00
	10.00	3.00	12.00	2.73	11.63	3.25
	828.38	589.86	1217.73	967.75	1382.42	1129.19
MACS3	1800.00	1800.00	1800.00	1800.00	1800.00	1800.00
	10.00	3.00	11.92	2.73	11.50	3.25
GA4	828.48	589.93	1221.10	975.43	1389.89	1159.37
	10.10	3.25	13.20	5.00	13.50	5.00
HGA5	861.00	619.00	1227.00	980.00	1427.00	1223.00
	10.00	3.30	13.10	4.60	12.70	5.60
SATS6	841.92	612.75	1213.16	952.30	1415.62	1120.37
	10.00	3.00	12.08	2.91	11.75	3.25
	828.38	589.86	1215.14	953.43	1385.47	1142.48
TESA7	201.00	1220.00	1474.00	3882.00	916.00	2669.00
	10.00	3.00	12.83	3.09	12.50	3.75
	828.76	589.86	1203.56	932.23	1363.84	1079.81
IACS-SA8	239.00	363.00	425.00	437.00	403.00	370.00
	10.00	3.13	13.33	3.18	12.88	3.38
RACS-VRPTW	960.94	678.81	1481.18	1215.12	1659.53	1427.19
	10.00	3.00	13.83	3.82	12.63	4.50
	838.12	591.13	1259.19	980.98	1436.58	1141.63
Our Heuristic	1663.89	2407.80	2029.99	1623.50	618.21	1819.69

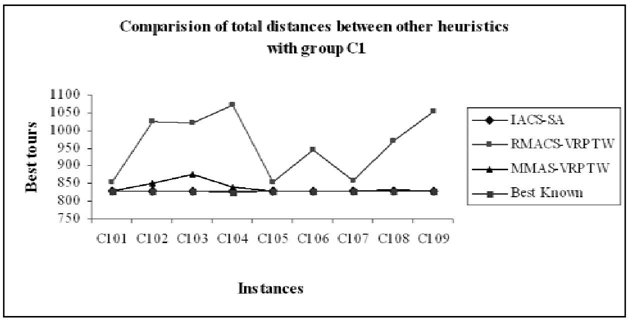

Fig. 12 *Comparison of MMAS-VRPTW with C1*

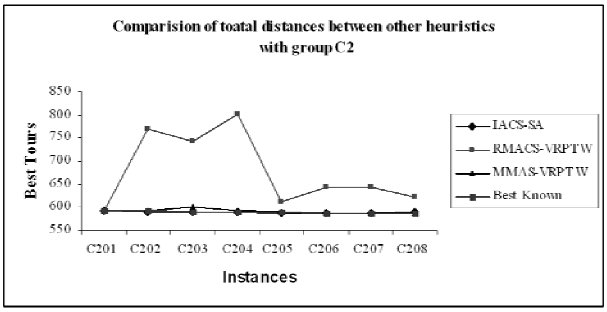

Fig. 13 *Comparison of MMAS-VRPTW with C2*

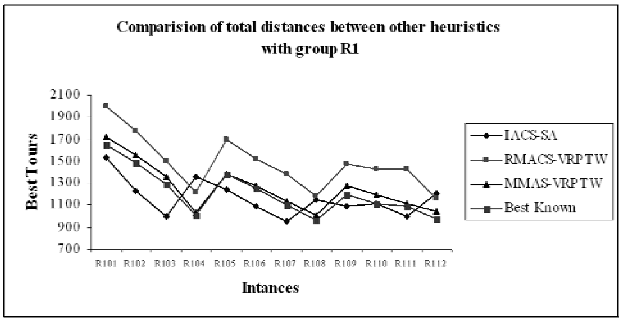

Fig. 14 *Comparison of MMAS-VRPTW with R1*

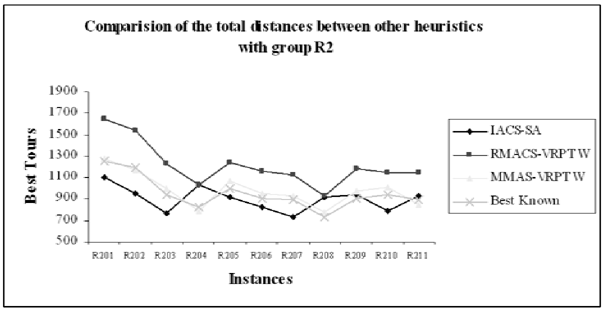

Fig. 15 *Comparison of MMAS-VRPTW with R2*

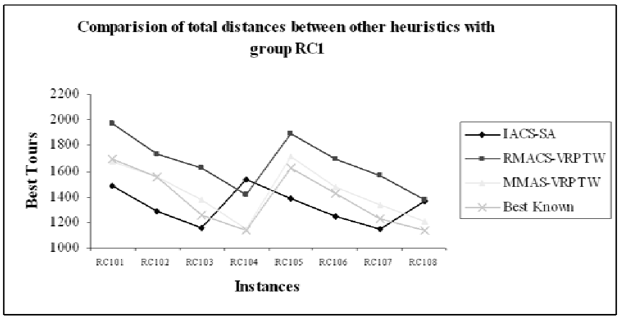

Fig. 16 *Comparison of MMAS-VRPTW with RC1*

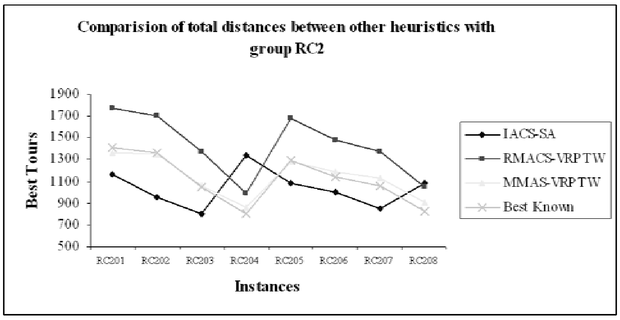

Fig. 17 *Comparison of MMAS-VRPTW with RC2*

7. Conclusion

When comparing gaps of the total travelled distance of all algorithms in Table 2, computational result has shown the viability of the MMAS approach to generate very high quality solution for the VRP and proposed method outperforms these competing heuristics in the terms of total travel distances. For in Fig. 12-17, an average performances gap about 2.702% (over all instances) of the best known results. Move-Exchanges and Hybrid 2-opt/*Or-Opt algorithms is the most powerful local search in this research which can improve the solution both inter-route and intra-route. However, it increases the computational time when every two or three swaps are examined. For the effects of computer performance are influence by many factors such as CPU speed, memory capacity, operation system and coding programming. Therefore, a fair transformation of computational time is difficult to establish. However, proposed method takes about 50 seconds to run for each problem instances, so it computational time is reasonable time. In this research, we are applying MMAS for solving VRPTW. The main idea is a modified ACO algorithm can efficiently to find the good solutions, but finding the optimal solution is still not guaranteed. The results may be because the solution space is too large and it can be easily by a local optimum. This problem is usually solved by employing in MMAS with combine local search techniques. Our results demonstrate that MMAS-VRPTW achieves a strongly improved performance compared to the other meta-heuristic for the VRPTW, which can be downloaded from the OR-library. The solution to this method have the potential to computational time is reasonable time and very high quality solution for the VRPTW.

Reference

Berger, J., Barkaoui, M. and Bräysy, O. (2001) A parallel hybrid genetic algorithm for the vehicle routing problem with time windows, Working paper, Defense Research Establishment Valcartier, Canada.

Chen, C.H and. Ting, C.J. (2004) An improved ant colony system algorithm for the vehicle routing problem, Working Paper 2004-001,

Chiang, W.C. and Russell, R.A. (1997) A reactive tabu search metaheuristics for the vehicle routing problem with time windows, INFORMS Journal on Computing, Vol. 9, 417-430.

Czech, Z.J. and Czarnas, P. (2002) A parallel simulated annealing for the vehicle routing problem with time windows, Proceedings 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, 376-383.

Desrochers, M., Desrosiers, J. and Solomon, M.M. (1992) A new optimization algorithm for the vehicle routing problem with time windows, Operations Research, Vol. 40, 342-354.

Gambardella, L.M., Taillard, E. and Agazzi, G. (1999) MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, 63-76, McGraw-Hill, London.

Lin, S. (1965) Computer solution of the traveling salesman problem, Bell System Computer Journal, Vol. 44, 2245-2269.

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

Solomon, M.M. (1987) Algorithms for the vehicle routing and scheduling problems with time window constrains, Operational Research, Vol. 35, No. 2, 254-265.

Stützle, T. and Dorigo, M. (1999) ACO algorithms for the quadratic assignment problem, In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization, McGraw-Hill.

Taillard, E., Badeau, P., Gendreau, M., Geurtin, F. and Potvin, J.Y. (1997) A tabu search heuristic for the vehicle routing problem with time windows, Transportation Science, Vol. 31, 170-186.

Tan, K.C., Lee, L.H. and Ou, K. (2001) Artificial intelligence heuristics in solving vehicle routing problems with time window constraints, Engineering Applications of Artificial, Vol. 14, 825-837.

Thangiah, S.R., Potvin, J.Y. and Sun, T. (1996) Heuristics approaches to vehicle routing with backhauls and time windows, Computers and Operations Research, Vol. 23, 1043-1057.

Ting, C.J. and Huang, C.H. (2005) An improved genetic algorithm for vehicle routing problem with time windows, International Journal of Industrial Engineering, 216–226.

Table 3 *The computational results*

	MMAS-VRPTW				Best Known			MMAS-VRPTW				Best Known	
	TD	NV	RPD	CPU Time	TD	NV		TD	NV	RPD	CPU Time	TD	NV
C101	828.940	10	0.00	2104.50	828.94	10	C201	591.560	3	0.00	2594.50	591.56	3
C102	850.250	10	2.57	990.65	828.94	10	C202	591.560	3	0.00	2435.58	591.56	3
C103	875.350	10	5.71	939.30	828.06	10	C203	600.200	3	1.53	2253.60	591.17	3
C104	840.550	10	1.91	958.62	824.78	10	C204	591.560	3	0.16	2356.25	590.60	3
C105	828.940	10	0.00	659.66	828.94	10	C205	588.880	3	0.00	2293.50	588.88	3
C106	828.940	10	0.00	2985.21	828.94	10	C206	588.490	3	0.00	2516.32	588.49	3
C107	828.940	10	0.00	2230.54	828.94	10	C207	588.270	3	0.00	2458.63	588.29	3
C108	832.250	10	0.40	2140.95	828.94	10	C208	588.490	3	0.03	2354.00	588.32	3
C109	828.940	10	0.00	1965.54	828.94	10	Avg.	591.126	3.00	0.21	2407.80	589.86	3
Avg.	838.122	10.00	1.18	1663.88	828.31	10							
R101	1715.750	20	4.25	410.39	1645.79	19	R201	1276.100	5	1.89	1810.83	1252.37	4
R102	1556.110	18	4.72	244.50	1486.02	17	R202	1169.190	5	-1.89	1523.50	1191.70	3
R103	1362.123	14	5.37	664.27	1292.68	13	R203	1001.370	4	6.58	1452.89	939.54	3
R104	1040.797	12	3.33	724.39	1007.24	9	R204	787.421	4	-4.62	1524.30	825.52	2
R105	1381.082	15	0.29	489.53	1377.11	14	R205	1068.750	4	7.47	1525.65	994.42	3
R106	1274.007	14	1.76	614.84	1251.98	12	R206	952.540	3	5.12	1432.45	906.14	3
R107	1137.264	13	2.95	735.63	1104.66	10	R207	923.024	3	3.32	1861.32	893.33	2
R108	1013.792	10	5.51	861.19	960.88	9	R208	778.429	3	7.11	1652.31	726.75	2
R109	1272.949	15	6.55	656.66	1194.73	11	R209	975.093	4	7.25	1765.52	909.16	3
R110	1196.705	13	6.98	727.13	1118.59	10	R210	1007.770	4	7.28	1654.95	939.34	3
R111	1113.163	11	1.50	765.49	1096.72	10	R211	851.125	3	-4.66	1654.78	892.71	2
R112	1046.485	11	6.55	826.98	982.14	9	Avg.	980.983	3.82	3.17	1623.50	951.91	2.73
Avg.	1259.186	13.83		2029.99	1209.89	11.92							
RC101	1675.350	15	-1.27	512.03	1696.94	14	RC201	1361.040	6	-3.26	1330.72	1406.91	4
RC102	1554.750	12	0.00	618.34	1554.75	12	RC202	1347.450	5	-1.44	1984.81	1367.09	3
RC103	1372.658	11	8.80	632.38	1261.67	11	RC203	1056.620	5	0.67	1854.78	1049.62	3
RC104	1152.578	11	1.51	765.06	1135.48	10	RC204	866.450	3	8.52	1986.65	798.41	3
RC105	1711.973	15	5.07	341.13	1629.44	13	RC205	1278.680	6	-1.43	1784.25	1297.19	4
RC106	1472.953	13	3.38	625.63	1424.73	11	RC206	1187.300	4	3.57	1985.15	1146.32	3
RC107	1341.705	12	9.04	704.13	1230.48	11	RC207	1126.950	4	6.20	1785.50	1061.14	3
RC108	1210.701	12	6.22	747.00	1139.82	10	RC208	908.516	3	9.71	1845.65	828.14	3
Avg.	1436.584	12.63	4.09	618.21	1384.16	11.5	Avg.	1141.626	4.5	2.82	1819.69	1119.35	3.25