A Max-Min Ant System Applied to The Vehicle Routing Problems

Dr. Suphan Sodsoon[1] and Asst. Prof.Dr. Sombut Sindhuchao[2]

Received: August 2008; Accepted: October 2008

Abstract

This work introduces a modified MAX-MIN Ant System (MMAS) algorithm to solve the Vehicle Routing problem (VRP), in which customers of known demand are supplied from a single depot. Vehicle Routing Problem is an NP-complete optimization problem and has usually been solved to nearly optimum by heuristics. The objective of VRP is to use a fleet of vehicles with specified capacity to serve a number of customers with dissimilar demands at minimum cost, without violating the capacity and route length constraints. Many meta-heuristic approaches like Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS) and An Improved Ant Colony System (IACS) algorithm. In this research, we proposed a Max-Min Ant System algorithm with local search approaches. Experiments on various aspects of 14 problem benchmark problems are other meta-heuristic and show that our results are competitive.

Keywords : Vehicle routing problem, Combinatorial optimization, Meta-heuristic, Ant Colony Optimization, Max-Min Ant System

1. Introduction

The basic vehicle routing problem (VRP) consists of a number customers, each requiring a specified demand of goods to be delivered. Vehicles dispatched from a single depot must deliver the goods required, and then return to the depot. Each vehicle can carry a limited demand and may also be restricted in the total distances its can travel. Only one vehicle is allowed to visit each customer. The problem is to find a set of delivery routes satisfying these requirements and giving minimal total cost. The VRP is a well-known NP-hard problem (Gambardella, L. M., et al. 1999) that is very difficult to solve to optimality. Exact methods like Dynamic Programming and Branch and Bound cannot obtain the optimal solution for large VRP within reasonable time, thus, many researchers have used heuristic approaches to solve the VRP. Many meta-heuristic approaches developed according to artificial intelligence, biological evolution and/or physics phenomenon have been reported and applied to the VRP, such as Simulated Annealing (SA) (Alfa, A. S., et al, 1991) Genetic Algorithms (GA) (Baker, B. M. and M., 2003) A., Tabu Search (TS) (Gendreau, M.,

[1]*Industrial Technology Rajamangala University of Technology Isan., Kasetsomboon Rd. Muang Kalasin, Thailand 46000 Tel: 0-4381-1128 Fax: 0-4381-3170 Email: suphan_sodsoon@hotmail.com*
[2]*Industrial Engineering, Ubonratchathani University., 85 Sathollmark Rd. Warinchamrap Ubonratchathani, Thailand 34190 Tel: 0-4528-8400-3 Fax: 0-4528-8378 Email: somsin@rocketmail.com*

(Gambardella, L., 1999). Among these meta-heuristic approaches, Ant System (AS) is a new distributed meta-heuristic first introduced by Coloni et al. (1991). The AS is based on the behavior of real ants searching for food. Real ants communicate with each other using an aromatic essence, called pheromone, which they laid down on the path they traversed. The pheromone will accumulates when more and more ants pass through the same path. Nevertheless, the pheromone will evaporate if no ants continue to pass. The selection of the pheromone trail reflects the length of the paths as well as the quality of the food source found. Dorigo et al. (1997a) reported the AS to solve traveling salesman problem (TSP), quadratic assignment problem (QAP) and job-shop scheduling. Dorigo and Gambardella Dorigo et al. (1997a) developed the ant colony system (ACS) to improve the performance of AS. Bullnheimer et al. (1999a) were the first researchers that used AS to solve the VRP. They presented a hybrid Ant System algorithm (HAS) that added the 2-opt heuristic and then based on saving algorithm to construct routes. However, the results of HAS were not as well as other meta-heuristic approaches. Then, Bullnheimer et al. (1999a) developed an improved AS (IAS) for the VRP. They applied the idea of candidate lists (Dorigo, M. and L. M. Gambardella, 1997a) to construct vehicle routes. Candidate lists can concentrate the search on promising nodes thus saving computational effort that can be better used for further iterations. Results of a set of standard problems showed that IAS was significantly better than AS and outperformed SA and Neural Network. Gambardella et al. (1999a) defined a hybrid Ant System algorithm for the VRP, which was inspired by ACS. Results obtained by HAS-VRP were competitive with those of the best-known algorithms and new upper bounds had been found for well-known problem instances. In this research, the main idea of our approach is to the transfer the sequential approach and parallel approach using a MMAS. We propose the heuristic to assign customers to each depot and constructing vehicle routes simultaneously. They are organized as follows. The model is formulated in section 2. In section 3 the MMAS and a solution improvement are presented. Computational experiments are discussed in section 4 and finally conclusion is provided in section 5.

2. Vehicle Rouing Problem

The vehicle routing problem is a very complicated combinatorial optimization problem that has been worked on since the late fifties, because of its central meaning in distribution management. The vehicle routing problem can be described as follows Gambardella, L. M., et al (1999) n customers must be served from a depot. Each customer asks v for a quantity q_i of goods. A fleet of vehicles, each vehicle with a capacity Q, is available to deliver goods. A service time t_i is associated with each customer. It represents the time required to service him/her. Therefore, a VRP solution is a collection of tours. The VRP can be modeled in mathematical terms through a complete weighted digraph $G = (V,A)$, where $V = \{0,1..,n\}$, is a set of nodes representing the depot (0) and the customers $\{0,1..,n)$ and $A = \{(i;j) | i, j \in V\}$ is a set of arcs, each one with a minimum travel time t_{ij} associated. The quantity of goods q_i requested by each customer i ($i > 0$) is associated with the corresponding vertex with a label. Labels $Q_1..,Q_2$ corresponding to vehicles capacities are finally associated with vertex 0 (the depot). A typical mathematical formulation for the single depot VRP is provided below:

Minimize: $$\sum_{i=0}^{N}\sum_{\substack{j=0 \\ j\neq i}}^{N}\sum_{k=1}^{K} c_{ij}x_{ij}^{k} \qquad (1)$$

Subject to $$\sum_{i=0}^{N}\sum_{k=1}^{K} x_{ij}^{k} = 1 \qquad \forall j \in \{1,...,N\} \qquad (2)$$

$$\sum_{j=0}^{N}\sum_{k=1}^{K} x_{ij}^{k} = 1 \qquad \forall j \in \{1,....,N\} \qquad (3)$$

$$\sum_{i=0}^{N} x_{ip}^{k} - \sum_{j=0}^{N} x_{pj}^{k} = 0 \qquad \forall p \in \{1,....,N\}, k \in \{1,...,K\} \qquad (4)$$

$$\sum_{j=0}^{N} q_i (\sum_{i=0}^{N} x_{ij}^{k}) \le Q \qquad \forall k \in \{1,....,K\} \qquad (5)$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} t_{ij}x_{ij}^{k} \le D \qquad \forall k \in \{1,....,K\} \qquad (6)$$

$$\sum_{j=1}^{N} x_{ij}^{k} \le 1 \qquad \forall k \in \{1,....,K\} \qquad (7)$$

$$\sum_{i=1}^{N} x_{ij}^{k} \le 1 \qquad \forall k \in \{1,....,K\} \qquad (8)$$

$$x_{ij}^{k} \in \{0,1\} \qquad \forall i,j \in \{1,...,N\}, k \in \{1,....,K\} \qquad (9)$$

The object function of distance minimization is expressed by Eq. (1). Constraints Eq. (2),(3) route continuity is enforce by (4) as once a vehicle arrived at a node, it must also leave that node. No vehicle can service customer demands that exceed the vehicle capacity in Eq. (5). A maximum route length is limited by Eq. (6). Eq.(7) and (8) ensures that each vehicle is scheduled no more than once and the last one is binary variable.

3. Appling Max-Min Ant System to The Vehicle Routing Program

In this research, we present MAX-MIN Ant System (MMAS) that it developed by Stutzle, T. & Hoos (1993). Our algorithm 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 can easily be extended by adding local search algorithms. MAX-MIN Ant System, which has been specifically developed to meet these requirements, differs in tree key aspects from AS.

a) To exploit the best solutions found during iteration or during the run of the algorithm, after iteration only one single ant adds pheromone. This ant may be the one which found the best solution in the current iteration, iteration best ant, or the one which found the best solution from the beginning of the trial, global best ant

b) To avoid stagnation of the search the range of possible trials on each solution component is limited to an interval. $[\tau_{min}, \tau_{max}]$.

c) We initialize the pheromone trials to τ_{max}, achieving in this way a higher exploration of solution at start of the algorithm.

3.1 The Proposed Heuristic Method

In our research, we applying MMAS for VRP include three steps are described as follows: The first steps, requires that a colony of ants is activated to find the shortest route by the procedure finds a feasible solution by an algorithm based on nearest neighbor and determine the initialized a number of vehicles (n_v), capable to cope with all emand. (n_v). To simply determine by the total demand divided into vehicle capacity. We used the amount of ant colonies equal to the number of vehicles plus one (n_v + 1)to construct routes according to Eq (10)

$$numberofvehicle\ (n_v) = \frac{Total\ demand\ of\ customers}{maximum\ load\ of\ vehicle} \qquad (10)$$

The second, an ant constructs routes using as multi colonies. In every generation, each ant *k* constructs one feasible solution by starting at the depot and successively choosing a next node or customer from the set of feasible nodes. An ant 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 in Eq. (14). Finally, after an ant has constructed its solution, we apply a local search to improve the solution quality. In particular, we apply 2-Opt and Or-Opt by exchanging a customer of route with a customer of another route.

Algorithm 1: MMAS-VRP algorithm

/*Main Procedure */

Step 1. /* Initialization *//* ω^{gb} is the best current solution,
$|N|$ is the number of nodes of the graph,
L_{NN} is the total travel time of the tour obtained by nearest neighbor,
$L_{\omega^{gb}}$ is the total travel time of the best tour found */ **Initialize variables Initialize** ω^{gb}
Start nearest neighbor $\tau_0 \leftarrow 1/(N * Z_0)$ $\tau_0 \leftarrow \tau_{max}$ Where: $\tau_{min} = \tau_{max} / 2* |N|$
Step 2. /* Main Loop */
Repeat: For each ant k **Call Route Construction Procedure()**
/* ω^k, L^k are current solution and current travel distances */
If(#visited_customer(ω^k) > #visited_customers(ω^{gb}) or (#visited_customer(ω^{gb})))
=#visited_customer(ω^{gb}) and $L^k < L_{\omega^{gb}}$ Then $\omega^{gb} := \omega^k$ End if End for
If(#visited_customer(ω^{gb})=0) Then
Step 3. /* Local Search */
Local Search() End
Else **Local Search()**
Step 4. /* Global pheromone updating */
If(i, j)is an edge in the current best solution Then $\tau_{ij} := (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}$
Where: $\Delta\tau_{ij} \leftarrow (L_{\omega^{gb}})^{-1}$ Else $\tau_{ij} := (1-\rho)\tau_{ij}$
End if

Fig 1. MMMAS algorithm

3.2 Parameters and Intialize the Pheromone Trails Phases

In our research, the initial pheromone level of each edge is evaluated by Eq. (11). The pheromone level of each edge has lower (τ_{min}) and upper limits (τ_{max}) and initial pheromone (τ_0) were set as in Eq.(12)

$$\tau_0 \leftarrow 1/(N * Z_0) \quad (11)$$

$$\tau_0 \leftarrow \tau_{max} \quad (12)$$

While $[\tau_{min}, \tau_{max}/2*|N|$ and N is the number of nodes. MMAS activated the pheromone level is set equal to τ_0 on each edge where $\tau_0 \leftarrow 1/(N * Z_0)$ .Z_0 is length of the solution found with the nearest neighbor algorithm. MMAS imposes explicit limits τ_{min} and τ_{max} on the minimum and maximum pheromone trails after iteration one has to ensure that the pheromone trails the limits. The maximum pheromone trails (τ_{max}) is set to an estimate of the maximum value. To determine reason able values for τ_{min}, we use the following assumptions, $\tau_{ij}(t), \tau_{max} \leq \tau_{ij}(t) \leq \tau_{max}$. After iteration one has to ensure that the pheromone trails respects to the limits. If we have $\tau_{ij}(t) > \tau_{max}$, we set $\tau_{ij}(t) = \tau_{max}$ and if, $\tau_{ij}(t) < \tau_{min}$, we set $\tau_{ij}(t) = \tau_{max}$. Also note that by enforcing we set $\tau_{min} > 0$ and if $\eta_{ij} < \infty$ for all solution components, the probability of choosing a specific solution component is 0.

3.3 Routes Contructions Phases

For constructing phases, we used the amount of ant colonies equal to the number of vehicles plus one ($n_v + 1$) to construct routes. It is an extension of the algorithm in Dang and Anulark, (2000) for ant colony construct routes in two frameworks between sequential and parallel constructions. These algorithms are illustrated in Figs. 2 and 3, respectively.

The sequential route construction, a route for a vehicle is constructed one at a time. When either the number of constructed routes or the total capacity a vehicle spent has reached the maximum number allowed a new route is initiated. The sequential construction phase ends if all gather points have been assigned to vehicles. On the other hand, in the parallel route construction illustrated in Fig 4. the first route of the parallel method is constructed for every vehicle at the same time. However, each tour must be not violated conditions of any routes or vehicles.

The parallel construction terminates when there is no more demand left. For an ant construction routes, after we know the number of multi-colonies and set them to positions on each vehicle. The multi 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 to *i* the next

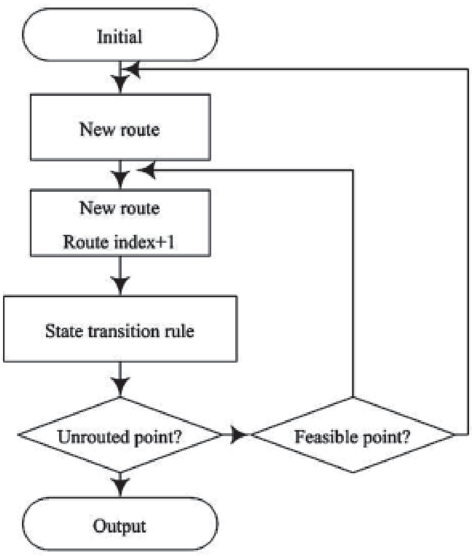


Fig 2. Sequential route construction

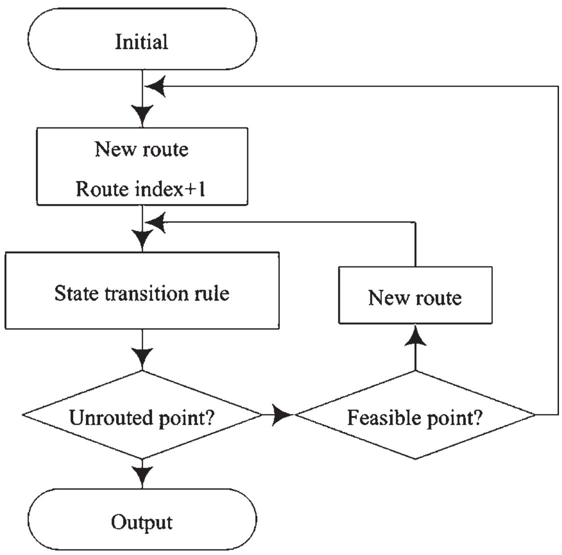


Fig 3. Parallel route construction

node *j* according to the state transition rule given by (13).

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]\left[\eta_{ij}\right]^{\beta}}{\sum_{j\in N_i^k}\left[\tau_{ij}(t)\right]\left[\eta_{il}\right]^{\beta}} \quad \text{if } j \in N_i^k \qquad (13)$$

Where α, β are two parameters which determine the relative importance of the pheromone trails and the heuristic information and N^k_i are the set of nodes that remain to be visited by an ant positioned on node *i*, $\tau_{ij}(t)$ is pheromone level on edges *(i.j)*, n_{ij} is the inverse of the length of edges (*i,j*) Thus $n_{ij} = 1/d_{ij}$,where d_{ij} is denoted the distance between nodes *i* and nodes *j*. In principle of MMAS algorithm can be applied to solve the VRP by defining solution components which the ants use to iteratively construct candidate solutions and on which they may deposit pheromone.

The artificial ants construct vehicle routes by successively choosing cities to visit, until each city has been visited. Whenever the choice of another city would lead to an infeasible solution for reasons of vehicle capacity or total route length, the depot is chosen and a new tour is started. At each step, every ant k computes a set of feasible expansions to its current partial solution and selects one of these probabilistically, according to a probability distribution specified.

maintaining solution feasibility. In this phase, we employ three local search procedures, namely Move-Exchanges, Or-opt and 2-opt algorithm which have been popular among exchange techniques proposed for solving VRP (Taillard et al., 1993, Potvin and Rousseau., 1995) The Overall improvement procedure of local search in MMAS is as follows Fig 5-7.

Type I. Move-Exchange modification

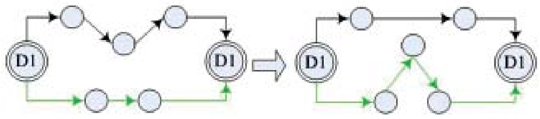


Fig 5. Move-Exchange algorithm

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.

Type II. 2-opt algorithm

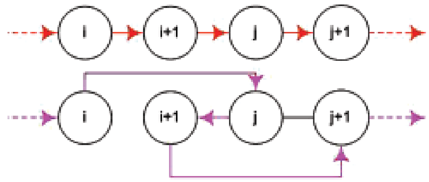


Fig 6. 2-opt algorithm

Algorithm 2: k-th ant route construction

/* k-th ant route construction */ Initialize ω^{k}

Repeat 1./* Definition of compatible customers */ For each not visited node j

if total distances from route ≤ max allowed route & capacity constriants is satisfied

Then is compatible $\eta_{ij} = 1/d_{ij}$ End if End for If there are compatible customers Then

2. /* State transition rule */ Exploration Eq(14) Insert j in ω^{k} End if

Until no compatible customers are found

Fig 4. k-th ant route construction

3.4 Routes Improvement Phases

After an ant has constructed its solution, we apply a local search algorithm to improve the solution quality.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 route length spent, while

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 Fig 6.

Type III. Or-opt algorithm

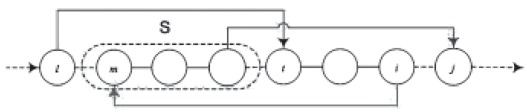


Fig. 7. Or-opt algorithm

The basic idea, suppose we want to move a string *S* of consecutive nodes immediately proceeding, respectively following Fig 7. Let *l* and *t* be the nodes immediately p For the further research, we will improve local search to obtain better solutions.Very large scale neighborhood search algorithm will also be studied to solve MDVRP receding, respectively following, *S* in the original route, let *i* and *j* be the nodes between whom *s* is to be inserted and let *m* and *k* be the first and the last nodes of *S*. Then:

$$MoveCost = d\ (l,t) + d\ (k+j) + d\ (i+m) - d\ (l,m) - d\ (k,t) - d(i,j\ \ ) \qquad (14)$$

3.5 Update of Pheromone Trails

The MMAS 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 $\tau_{min} \leq \tau_{ij} \leq \tau_{min}$. The pheromone trails are initialized to the upper trail limits. After all ants have constructed solutions, the pheromone trails are updated according to (15).

$$\tau_{ij}(t+1) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best} \qquad (15)$$

Where ρ a parameter is called evaporation coefficient, $0 < \rho < 1$ and $\triangle\ \tau_{ij}^{best} = 1/C^{best}$ 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 $\triangle\tau_{ij}^{best}$. If an edge *(i, j)* does not belong to the so far best solution, the intensity of pheromone will be reduced.

4. Number 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.

4.1 The Vehicle Routing Problem Instances

Table 1 contains the data for the 14 vehicle routing problem instances (Gambardella, L. M. et al., 1999). These problems contain between 50 and 199 customers as well as the depot.

4.2 Experment with MMAS

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

Table 1 : Vehicle routing problem instances

NO	N	Q	s/T	BKS	Reference	Method
C1	50	160	0/∞	524.61	Taillard(1993)	TS
C2	75	140	0/∞	835.26	Taillard(1993)	TS
C3	100	200	0/∞	826.14	Taillard(1993)	TS
C4	150	200	0/∞	1028.42	Taillard(1993)	TS
C5	199	200	0/∞	1291.45	Rochat and Taillard(1995)	TS
C6	50	160	10/200	555.43	Taillard(1993)	TS
C7	75	140	10/160	909.68	Taillard(1993)	TS
C8	100	200	10/230	865.94	Taillard(1993)	TS
C9	150	200	10/200	1162.55	Taillard(1993)	TS
C10	199	200	10/200	1395.85	Rochat and Taillard(1995)	TS
C11	120	200	0/∞	1042.11	Taillard(1993)	TS
C12	100	200	0/∞	819.56	Taillard(1993)	TS
C13	120	200	50/720	1541.14	Taillard(1993)	TS
C14	100	200	90/1040	866.37	Taillard(1993)	TS

N= number of nodes; Q=capacity of vehicle; s/T=route length constraints determines the convergence speed of MMAS towards a good solution. We present curves for the trade-off between the best solutions is found versus serial and parallel method on routes building for C1 instances by using difference setting of ρ is varies between 0.7 and 0.98, as shown in Fig 8.

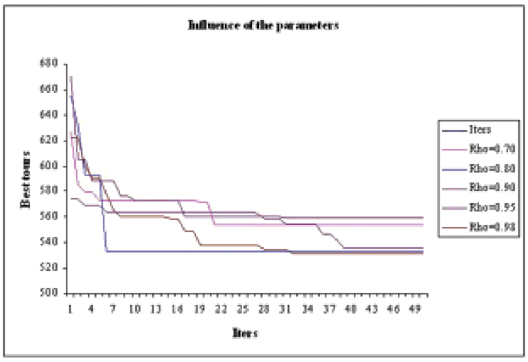
In Fig 8, it can be observed that the best tours are found when we using parallel method more than serial method and for a low number of iterations, better tours are found when using high values of ρ. This is due to the fact that for high ρ the pheromone trails on arcs which are reinforced decrease faster, and hence, the search concentrates earlier around the best tours seen so far. If ρ is low, too few iterations are performed to search marked differences between the pheromone trials on arcs contained in high quality solution and those which are not part of the best solutions.

Therefore, these parameters have a good performance at values around α =1, β = 2, ρ = 0.98 and the number of maximum iteration is n and solve 5 times for each problem, which are set for all experiments of this study.

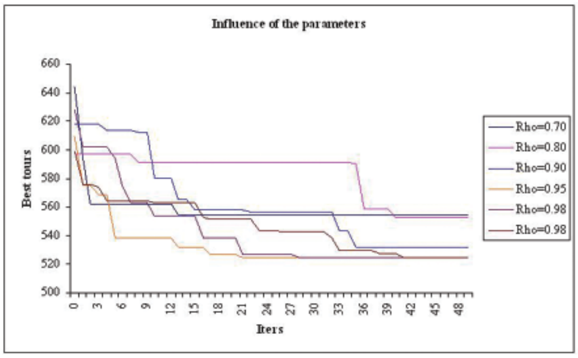
4.3 Computation Experiment

MMAS have been coded in the visual C++ and experiments were run on a Pentium IV, 256 MB of RAM, 3.07 GHz processor. In order to asses the relative performances of MMAS with local search independently from details of the settings, we used α =1, β = 2, ρ = 0.98. For all problems maximum iteration times are and n solve 5 times for each problem.

In Table 2 We show the overall of the experiment was compared the effect between serial and parallel method for conducted the best



Type I. Sequential Route Construction

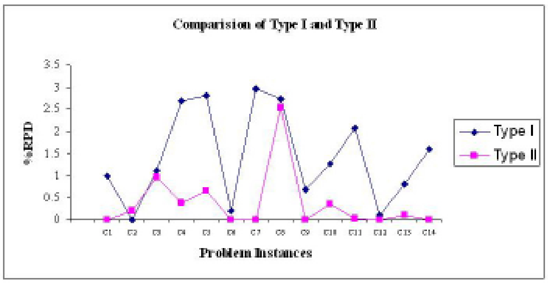


Type II. Parallel Route Construction

Fig 8 Influence of the parameters ρ on the trade-off the Type I and Type II

Table 2: The computational results of MMAS with local search algorithm

NO	BKS	BT-Type I	%RPD	Time (Sec.)	BT-Type II	%RPD	Time (Sec.)
C1	524.610	529.851	0.999	96.860	524.610	**0.000**	96.079
C2	835.260	835.260	**0.000**	1070.340	837.032	0.212	1028.671
C3	826.140	835.357	1.116	1060.780	834.063	0.959	1189.203
C4	1028.420	1056.110	2.692	2926.660	1032.400	0.387	2970.890
C5	1291.450	1327.780	2.813	891.330	1300.100	0.670	901.028
C6	555.430	556.616	0.214	86.200	555.430	**0.000**	79.641
C7	909.680	936.649	2.965	143.580	909.680	**0.000**	140.750
C8	865.940	889.550	2.727	230.170	887.914	2.538	230.547
C9	1162.550	1170.490	0.683	2935.660	1162.550	**0.000**	2924.657
C10	1395.850	1413.382	1.256	876.470	1400.730	0.350	870.690
C11	1042.110	1063.690	2.071	965.860	1042.250	0.013	652.634
C12	819.560	820.325	0.093	1025.160	819.560	**0.000**	1004.547
C13	1541.140	1553.593	0.808	1015.240	1542.864	0.112	1000.656
C14	866.370	880.285	1.606	960.500	866.370	**0.000**	949.658
Avg.	976.036	990.638	1.432	1020.344	979.682	0.374	941.926



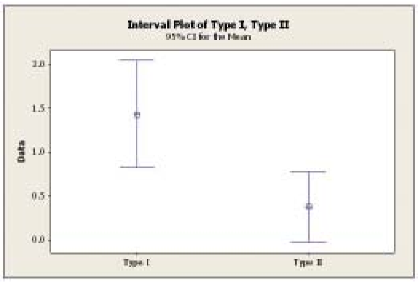


Fig 8 Comparison of construction solution methods

In that table, the following notation is used:
BT -Type I= Serial method
and BT -Type II= Parallel method
BT- Obj..= the best length tour of objective function
Avg. CPU Time (sec.) = Average computational time (sec.)
BKS = the best known solution from OR-library
(%)RPD = ((BT- Obj.-BKS)/BKS)*100%

solution. In Fig 8, the proposed heuristic can find some good solutions with reasonable time. The relative percentages deviation of the parallel method and serial method has been a range of varies are 0.0%-2.54% and 0.0%-2.81%, respectively.

4.4 Coparisons of MMAS with Other Meta-Heuristics

In this section, we will present numerical results for our approach and compare them with results of the other meta-heuristics in terms of RPD, shown in the Table 3, we compare the performance of the proposed method to that of SA TS GA and IACS, the numbers in bold indicate the best solution among eight algorithms. It can be observed from Table 3 that the proposed method is able to find the better solutions for test problem C1 C2 C6 C7 C9 C12 and C14, so it is equal to the best known solutions in the operation-library. Then, we can see the proposed method outperformed in the term total travel distances of all methods.

The MMAS also yields the best solutions among the other algorithms in 10 out of 14 problems, as shown in Table 3 but some instances inferior to IACS method like C3 C8 and C11 in term of RPD. However, the proposed method can provide a better solution than IACS method such as in C4 C5 C9 C10 C12 and C13

In Fig 9, an average performance gap about 0.36% 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 the computer performance are influenced 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, our method can take average computational time equal to 941.93 seconds for

Table 3 : Computational results of MMAS

NO	[1]AS	[2]IAS	[3]SA	[4]SA	[5]TS	[6]TS	[7]GA	[8]IACS	MMAS
C1	**0.00**	**0.00**	0.65	**0.00**	**0.00**	**0.00**	0.04	**0.00**	**0.000**
C2	4.23	1.08	0.40	0.69	0.06	**0.00**	1.74	**0.01**	**0.000**
C3	6.45	0.75	0.37	0.47	0.40	**0.00**	1.76	*0.61*	***0.959***
C4	11.57	3.22	2.88	3.36	0.75	0.11	2.67	0.84	**0.387**
C5	14.09	4.03	6.55	5.31	2.42	0.55	6.76	2.25	0.670
C6	1.35	0.87	**0.00**	**0.00**	**0.00**	**0.00**	0.87	**0.00**	**0.000**
C7	4.23	0.72	**0.00**	1.13	0.39	6.15	0.49	**0.00**	**0.000**
C8	2.34	0.09	0.09	0.47	0.00	1.75	0.79	0.00	*2.538*
C9	3.39	2.88	0.14	2.96	1.31	-	2.62	0.61	**0.000**
C10	7.80	4.00	1.58	4.74	1.62	3.11	6.25	0.99	**0.350**
C11	2.91	2.22	12.85	**0.00**	3.31	**0.00**	1.74	*0.00*	0.013
C12	0.05	**0.00**	0.79	0.18	**0.00**	**0.00**	7.11	1.50	**0.000**
C13	3.20	1.22	0.31	1.74	2.12	5.02	1.37	0.29	**0.112**
C14	0.40	0.08	2.73	0.07	**0.00**	5.64	0.69	**0.00**	**0.000**
Avg.	4.43	1.51	2.09	1.36	0.86	1.60	2.49	0.51	0.360

[1] AS by Bullnheimer et al. (1998) [2] IAS by Bullnheimer et al. (1999a) [3] SA by Osman (1993) [4] SA by Van Breedam (1995) [5] TS by Gendreau et al. (1994) [6] TS by Xu and Kelly (1996) [7] GA by Baker and Avechew (2003) [8] IACS by Ho Chan et al (2006)

each instances (or 15.69 minutes), so it's reasonable time.

5. Conclusions

In this research, we have proposed MMAS for solving VRP. It is efficiently to find the good solutions. In addition, we have evaluated the performance of our results with other heuristics such as Simulated Annealing (SA) Tabu Search (TS) Genetic Algorithm (GA) and Improved Ant Colony System (IACS) method. The MMAS method can be outperformed SA TS and GA, but in some cases inferior to IACS. The results indicate that this method performed as well in terms of the solution quality and run time consumed by compared with other heuristic approach on 14 test problems. Our results demonstrate that MMAS achieves a strongly performance for the VRP. For further research, we will improve local search to obtain better solutions. Very large scale neighborhood search algorithm will also be studied to solve VRP.

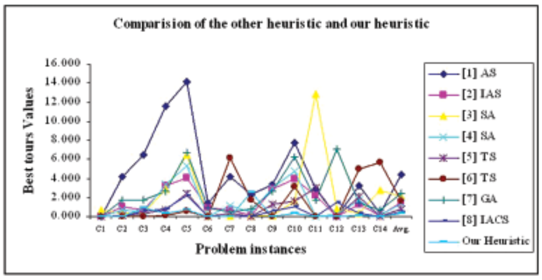


Fig 9 Comparison of the other heuristic and our heuristic

Ackhnowledgements

This work was partially financially supported by Rajamangala University of Technology Isan., Thailand.

Referrences

Alfa, A. S. et al., (1991). A 3-opt based simulated annealing algorithm for vehicle routing problems, **Computers & Industrial Engineering.** Vol.21 pp.635-639.

Baker, B. M. and Ayechew M. A. (2003). A genetic algorithm for the vehicle routing problem, **Computers & Operations Research.** Vol. 30 pp. 787-800.

Bullnheimer, B. et al., (1998). **Applying the ant system to the vehicle routing problem.** Boston.

Chia-Ho Chen and Ching-Jung Ting (2006). An improved ant colony system algorithm for vehicle routing problem. **Journal of the Chinese Institute of Industrial Engineers.** Vol. 23(2): 115-126.

Christofides, N. et al., (1979). The vehicle routing problem. in: N. Christofides, A. Mingozzi, P. Toth and C.Sandi (eds.), **Combina torial Optimization,** pp. 315-338. Wiley, Chichester.

Dang Vu Tung and Anulark Pinnoi (2000). Case Study: vehicle routing-scheduling for waste collection in Hanoi. **European Journal of Operational Research:** vol. 125, issue 3, pp.449-468.

Dorigo, M. and Gambardella L. M. (1997). Ant colonies for the traveling salesman problem. **Bio-Systems.** Vol. 43, pp.73-81.

Gambardella, L. M. et al., (1999). **Ant colonies for vehicle routing problems.** McGraw-Hill.

Gendreau, M. et al., (1994). A tabu search euristic for the vehicle routing problem. **Management science:** vol.40(10),pp.1276-1290.

Golden, B. L. et al., (1998). Meta-heuristic in vehicle routing. In T. G. Crainic G. Laporte, editors. Fleet **Management and Logistics,** luwer, Boston.

Osman, I. H. (1993). Meta strategy simulated annealing and tabu search algorithms for the vehicle routing problem. **Annals of Operations Research.** Vol 4 , pp. 421-451.

Potvin, J.K. and Rousseau, J.M. (1995). An exchange heuristic for routing problem with time windows. **Journal of the Operational Research Society.** Vol.46, pp. 1433-1446.

Rochat, Y. and Taillard E. (1995). Probabilistic and intensification in local search for vehicle routing. **Journal of Heuristics.** Vol, 1, pp 147-167.

Stützle, T. and M. Dorigo (1999). **ACO algorithms for the quadratic assignment problem.** McGraw-Hill,

Taillard, E. (1993). Parallel iterative search methods for vehicle routing problems. **Networks An International Journal.** Vol23 Issue 8, pp.661-673.