Proceedings of the 21st IASTED International Conference
APPLIED INFORMATICS
February 10-13, 2003, Innsbruck, Austria

A MULTIOBJECTIVE ANT COLONY SYSTEM FOR VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Benjamín Barán and Matilde Schaerer
Centro Nacional de Computación, Universidad Nacional de Asunción
San Lorenzo, P.O. Box 1439
Paraguay
bbaran@cnc.una.py, schma@rieder.net.py

ABSTRACT

This paper proposes a variation of the Multiple Ant Colony System for Vehicle Routing Problem with Time Windows (MACS-VRPTW) algorithm, which is based on an Ant Colony System approach, using two ant colonies to minimize first the number of vehicles and then the total traveled distance. As an improvement, the present work proposes to use a modified version specialized for a multiobjective context, using just one colony to get a set of Pareto optimal solutions considering three objectives at the same time, the number of vehicles, the total traveling time and the total delivery time. Experimental results validate the new approach with very good results when compared to the original MACS-VRPTW.

KEY WORDS

Multiobjective optimization, Ant Colony Optimization, Vehicle Routing Problem with Time Windows, Pareto Optimal Set.

1. INTRODUCTION

The Vehicle Routing Problem with Time Windows (VRPTW) [1] is an extension of the Vehicle Routing Problem (VRP) [2], in which the aim is to find a set of minimum-cost vehicle routes, that originates and terminates at a central depot, for a fleet of vehicles that serve a given set of customers with known demand. Each customer is served exactly once and all the customers must be assigned to vehicles without exceeding vehicle capacities. VRPTW added to these issues the complexity of allowable delivery times, or time windows. With the time windows, the total routing and scheduling cost include not only the total travel distance and time costs, but also the cost of waiting time incurred when a vehicle arrives too early at a customer location. The service beginning time at each customer must be greater than or equal to the beginning of the time window, and the arrival time at each customer must be lower than or equal to the end of the time window. If a vehicle arrives to a customer before the beginning of the time window, it has to wait until this time to serve the customer.

The spatial problem of routing vehicles has been extensively studied in the literature. Several approaches have been published dealing with it, using parallel implementations [3], hybrid strategies coupling local search method to evolutionary algorithm [4], neuronal network [5, 6], a heuristic that uses pheromone information [7], and works that deal with this problem using evolutionary calculation to optimize the demand for each vehicle besides the optimization of the vehicle number and the total traveling time [8].

More recently, the time window constraint has been considered and many approaches have been presented to solve the VRPTW: parallel algorithms with a polynomial number of processor [9], genetic algorithm [10, 11, 12, 13], and parallel simulated annealing [14, 15].

In this context, MACS-VRPTW [16] was developed as an evolutionary proposal based on Ant Colony System (ACS), and more generally, on Ant Colony Optimization (ACO), with the aim of optimizing two objective functions, as summarized in section 3. In fact, ACO is a metaheuristic approach that imitates ants' behavior, where ants cooperate in their search for food by depositing chemical traces (pheromones). In a computational implementation, ants that found good solutions deposit pheromones in their paths. That way, ants of the following generations may decide with good probability to follow a good trace with greater quantity of pheromones.

This paper is concerned with solution construction for the VRPTW using ACS. The approach presented may be considered an extension of the MACS-VRPTW algorithm for a truly multiobjective context. The novelty of the presented approach consists in incorporating the Pareto optimal concept in such a way that the final solution is not a single optimum but a whole set of Pareto optimal solutions where all objectives are equally considered, i.e., a set of solutions where no solution can be improved in any objective without damaging other objectives [17].

The rest of this work is organized as follows: section 2 formalizes the problem, section 3 introduces the MACS-VRPTW while its improved version is presented in

section 4. Section 5 presents performance metrics used for comparison; while section 6 presents experimental results, with conclusions left for section 7.

2. THE PROBLEM

The Vehicle Routing Problem (VRP) with Time Windows, also known as VRPTW, is an extension of VRP that may be mathematically stated as:

Minimize an objective function F

given:

$C = \{c_0, c_1, c_2, ..., c_{\tilde{n}}\}$, … that represents the set of $\tilde{n}$ customers, where c_0 represents the depot and c_i a customer i;

t_{ij} … traveling time between c_i and c_j ($t_{ij} = t_{ji}$);

Q ... total capacity of a vehicle (assumed homogeneous);

q_i … demand of customer c_i where $q_i \leq Q$ and $q_0 = 0$;

$[b_i, e_i]$… acceptable time window for customer c_i, with b_i as the earliest and e_i as the latest time to serve c_i.

In a traditional VRPTW problem, a single objective function F may be chosen from any of the following ones:

Number of vehicles

$$F_1 = \nu \tag{1}$$

where ν represents the total number of vehicles.

Total traveling time (without considering waiting times)

$$F_2 = \sum t_{ij} \quad \forall\, t_{ij} \in \psi \tag{2}$$

where ψ represents a complete solution (or *tour*)

Total delivery time (considering waiting times)

$$F_3 = \sum_{i=1}^{\nu} T_i \tag{3}$$

where T_i represents the time needed by vehicle i to return to the depot c_0 considering the waiting time needed at each customer when it arrives before the beginning of the time window.

For the multiobjective context of the present work, the objective function F is considered as a three-dimensional vector, i.e.:

$$F = [F_1\ F_2\ F_3]^T$$

with no objective considered more important than the others.

3. MACS-VRPTW REVIEW

MACS-VRPTW has been designed to solve the vehicle routing problems with time windows using an ant colony system [16]. This approach uses two different ant colonies to minimize two objective functions. The first colony, denoted as ACS-VEI, minimizes the number of vehicles (F_1), while the second colony, denoted as ACS-TIME, minimizes the total traveling time (F_2). In this original approach, the first objective takes precedence over the second.

Both colonies use independent pheromone trails and collaborate sharing a global best solution, which is used for pheromone updating. Given a feasible solution ψ with ν vehicles, the MACS-VRPTW decreases one vehicle at a time, trying to find a feasible solution with (ν-1) vehicles, using its first ant colony (ACS-VEI) and a heuristic that maximize the number of visited customers with each vehicle. At the same time, the second colony (ACS-TIME) tries to minimize the total traveling time with the given number of vehicle (ν) of the global best solution ψ. When the first colony ACS-VEI finds a new solution with a fewer number of vehicle, both colonies are reinitialized and the older solution with a larger number of vehicles is forgotten, i.e., only one global solution is kept at a time. The pseudocode below shows this approach.

Pseudocode MACS-VRPTW

1. /* Initialization */
   /* ψ^{gb} is the best feasible solution: lowest number of vehicles and shortest traveling time.
   #active_vehicles (ψ) computes the number of active vehicles in the feasible solution ψ */
   $\psi^{gb} \leftarrow$ feasible initial solution with unlimited number of vehicles calculated with a nearest neighbor heuristic.
2. /* *Main loop* */

Repeat
  $\nu \leftarrow$ #active_vehicles(ψ^{gb})
  Activate ACS-VEI(ν - 1)
  Activate ACS-TIME(ν)
  while ACS-VEI and ACS-TIME are active
    wait for an improved solution ψ from ACS-VEI or ACS-TIME
    $\psi^{gb} \leftarrow \psi$
    if #active_vehicles(ψ^{gb}) < ν then
      reinitialize ACS-TIME and ACS-VEI
  end while
until a stopping criterion is met.

To construct a solution, the depot is duplicated a number of times equal to the number of available vehicles, and the distances between copies of the depots are set to zero. Each artificial ant starts from a randomly chosen copy of the depot and, at each step, moves to a not yet visited node that does not violate neither the time window constraints nor the vehicle capacity. The set of available customers also includes not yet visited duplicated depots. A feasible solution ψ is represented in Figure 1.

It should be noted that in general, the original MACS-VRPTW algorithm only finds one *optimal* solution with

the minimum v. It does not find a Pareto set of optimal solutions; therefore, it may not be considered a truly general multiobjective optimization algorithm [17]. Consequently, to compute a whole set of Pareto solutions, the original proposal was slightly modified to store all Pareto optimal solutions instead of erasing good solutions when a new one with fewer number of vehicles is found.

Considering that the original MACS-VRPTW was not designed for a truly multiobjective context, the following section presents a new approach that considers the multiobjective nature of the problem stated in section 2.

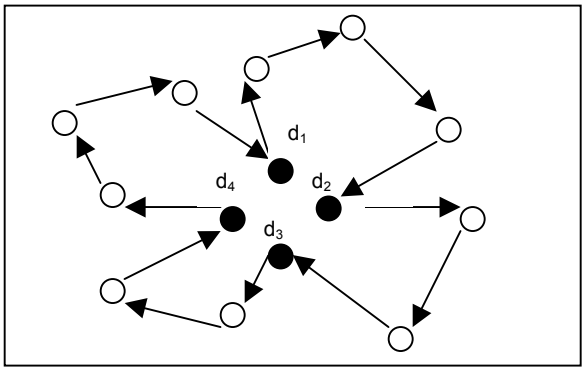

Fig. 1. Feasible solution for a vehicle routing problem with four duplicated depot, i.e. $v = 4$.

4. NEW MULTIOBJECTIVE APPROACH

This work presents a multiobjective vehicle routing problem with time windows that simultaneously optimize the three objective functions presented in section 2, i.e.:

- *Number of vehicles,*
- *Total traveling time,* and
- *Total delivery time.*

The main difference with the original MACS-VRPTW approach is to consider all the objectives with the same importance, i.e., no objective has any precedence over the others; therefore, a set of Pareto optimal solutions P may be found [18].

The proposed new approach uses an unique ant colony to simultaneously minimize all three functions. All objectives share the same pheromone trails. Therefore, the *knowledge* of good solutions is equally important for every objective function in the Pareto front.

The idea of the algorithm is to construct only feasible solutions using as many vehicles as needed. In every generation, each ant k (of a set of m ants) constructs one feasible solution, starting at the depot and successively choosing a next node or customer c_j, from the set of feasible nodes, N_i^k where subindex i represents that ant k is in node c_i. N_i^k is calculated at each node c_i, and includes all nodes not yet visited and that do not violate any constraint (as time window and vehicle capacity). This set N_i^k does not include depots. When a vehicle can not add more nodes, a depot is included, and another vehicle starts adding nodes in the same way. This process is repeated until all nodes have been visited and, therefore, a feasible solution ψ was found.

An ant k moves from node c_i to node c_j using heuristic as well as pheromone information. The heuristic information is given by the visibility, while the pheromone information, denoted by $\tau(i,j)$, indicates how well seems to visit customer c_j after c_i considering the solutions already found.

To choose the next node c_j to be visited by an ant k in c_i a probability $p_k(i,j)$ is assigned to each city and the next city c_j is chosen in N_i^k by the following procedure:

Procedure Choose-Next-Node
  Randomly choose to do exploration with probability q_0 or exploitation otherwise;
  if exploitation
    Choose the city with larger $p_k(i,j)$
  else (exploration)
    Randomly choose c_j using probabilities $p_k(i,j)$

where:

$$p_k(\mathrm{i,j}) = \begin{cases} \dfrac{[\tau(i,j)][\eta_L(i,j)]^{\lambda\beta} \cdot [\eta_J(i,j)]^{(1-\lambda)\beta}}{\sum_u [\tau(i,u)][\eta_L(i,u)]^{\lambda\beta} \cdot [\eta_J(i,u)]^{(1-\lambda)\beta}} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$\lambda = k/m$, and β weights the relative importance of the objectives with respect to the pheromone trail, given by τ.

The visibility for the objective function F_2 considering traveling time is calculated as:

$$\eta_J(i,j) = 1 / \mathrm{t_{ij}} \quad (5)$$

while the visibility for objective function F_3 is related to the waiting time and the width of the time windows. It is calculated as:

$$\begin{aligned} \mathrm{ct}_j &= \max(ct_i + t_{ij};\, b_j) \\ \Delta t_{ij} &= \mathrm{ct}_j - ct_i \\ d_{ij} &= \Delta t_{ij} \,.\, (e_j - ct_i) \\ d_{ij} &= \max(1,\, d_{ij}) \\ \eta_L(i,j) &= 1 / d_{ij} \end{aligned} \quad (6)$$

with ct_j representing the current time at node j (or delivery time to c_j). Each vehicle begins its trip at a depot c_0 with $ct_0 = 0$.

When each ant k finds a complete solution ψ^k, it is compared to the Pareto optimal set P to check if it is non-dominated (worse solutions are dominated). If it is a new optimal Pareto solution, it is included in P and dominated solutions from P are erased. At the end of each

generation, τ_0' is calculated with the average values of the Pareto optimal set as follows:

$$\tau_0' = 1 / n.\overline{L_\psi^P}.\overline{J_\psi^P} \quad (7)$$

where :

n… $\overline{v_\psi^P} + \tilde{n}$ (customers) is the average number of nodes;

$\overline{v_\psi^P}$ … is the average vehicle number;

$\overline{L_\psi^P}$ … is the average delivery time;

$\overline{J_\psi^P}$ … is the average total time.

If $\tau_0' > \tau_0$ because the Pareto set P was improved, the pheromone trail is reinitialized with the new value τ_0' to improve exploration; otherwise, exploitation is favored by globally updating the pheromone trail with each solution of the current Pareto optimal set P, using the following equation:

$$\tau(i,j) = (1 - \rho).\tau(i,j) + \rho / (L_\psi^P.J_\psi^P) \quad \forall (i,j) \in \psi^P. \quad (8)$$

where L_ψ^P is the value of F_3 for a given solution ψ used to update $\tau(i,j)$ while J_ψ^P is the corresponding value of F_2.

The re-initialization of $\tau(i,j)$ forces the ants to explore new ways, without the probably wrong information that has been introduced by solutions that were already erased from P because they were dominated by new ones. This exploration is powered by an evaporation process that occurs when an ant moves from node i to node j. In this case, $\tau(i,j)$ is updated according to:

$$\tau(i,j) = (1 - \rho) . \tau(i,j) + \rho. \tau_0. \quad (9)$$

where τ_0 is initially calculated, in a similar way of equation (7), according to:

$$\tau_0 = 1 / (n . L_\psi^h . J_\psi^h)$$

with L_ψ^h representing an initial estimation of the total delivery time (F_3), while J_ψ^h represents an initial estimation of the total traveling time (F_2), and n is the initial number of nodes (depots + customers). The pseudocode for the new proposal denoted as MOACS-VRPTW (Multi-Objective Ant Colony System for the Vehicle Routing Problem with Time Windows) follows:

Pseudocode MOACS-VRPTW
/* *Initialization* */
  $\psi^h \leftarrow$ feasible initial solution
  $\tau_0 = 1 / n.J_\psi^h. L_\psi^h$
Repeat /* *Main Loop* */
  for each ant $k \in \{1,\ldots, m\}$ **do**
    /* Construct a solution (ψ^k) */
    $\psi^k = build\_tour(k)$
    if $\psi^k \in P$
      save ψ^k and erase dominated solution from P
    end if
  end for
  /* Calculate τ_0' according to (7)*/
  $\tau_0' = 1 / n.\overline{L_\psi^P}.\overline{J_\psi^P}$
  if $\tau_0' > \tau_0$
    /* A better Pareto set P was found */
    $\tau_0 \leftarrow \tau_0'$
    Re-initialize trails $\{\tau\}$ with τ_0
  else
    for each $\psi^P \in$ P **do**
      /* perform global updating according to (8) */
      $\tau(i,j) = (1 - \rho) . \tau(i,j) + \rho / L_\psi^P.J_\psi^P \quad \forall (i,j) \in \psi^P$
    end for
  end if
until a stopping criterion is met.

Procedure *build_tour*(k)
/* *Initialization* */
  Put ant k at depot 0 (<i>)
  $\psi^k \leftarrow$ <i>
  $ct_k \leftarrow 0$, $load_k \leftarrow 0$
/* The ant builds its tour. Tour is stored in ψ^k */
repeat
  /* Starting from node i compute the set N_i^k of feasible nodes*/
  compute N_i^k
  if $N_i^k = \{\}$ /* no feasible nodes */
    $ct_k \leftarrow 0$
    $load_k \leftarrow 0$
    $\psi^k \leftarrow$ <i>
  else
    /* $\forall$ j $\in N_i^k$ compute visibility */
    for every j $\in N_i^k$ **do**
      Compute $\eta_J(i,j)$ using equation (5)
      Compute $\eta_L(i,j)$ using equation (6)
    end for
    <j> $\leftarrow$ Choose-Next-Node
    $\psi^k \leftarrow \psi^k \cup$ <j>
    $ct_k \leftarrow \max(ct_i + t_{ij}; b_j)$
    $load_k \leftarrow load_k \cup q_j$
    /* Pheromone updating using equation (9) */
    $\tau(i,j) = (1 - \rho) . \tau(i,j) + \rho. \tau_0$
    $i \leftarrow j$
  end if
until all customers have been visited.

5. PERFORMANCE METRICS

To evaluate experimental results using the two versions described in the preceding sections, an appropriate test suit of metrics was chosen from [17], considering that no single metric can entirely capture performance, effectiveness and efficiency for multiobjective evolutionary algorithms. The suit comprises the following metrics:

1) Overall Non-dominated Vector Generation (*ONVG*): simply counts the number of solutions in the calculated Pareto Front, denoted as $\boldsymbol{Y}_{known}$

$$ONVG \stackrel{\Delta}{=} |\, Y_{known} \,|_c \qquad (10)$$

where $|\ |_c$ denotes cardinality. The larger the value of the ONVG, the better for knowing Pareto Front details.

2) Overall True Non-dominated Vector Generation (*OTNVG*): counts the number of solutions in $\boldsymbol{Y}_{known}$ that are also in the True Pareto optimal Front denoted as $\boldsymbol{Y}_{true}$. Because $\boldsymbol{Y}_{true}$ is not known in theory, it may be estimated running several different evolutionary algorithms for a large number of times, for the same problem, choosing the optimal Pareto solutions found in all the experiments. For a good approximation of $\boldsymbol{Y}_{true}$ a large number of running using as many algorithms as possible are needed. Clearly, the larger is the OTNVG, the better is a solution set $\boldsymbol{Y}_{known}$.

$$OTNVG \stackrel{\Delta}{=} \left| \left\{ y \mid y \in Y_{known} \wedge y \in Y_{true} \right\} \right|_c \qquad (11)$$

3) Overall Non-dominated Vector Generation Ratio (*ONVGR*): denotes the ratio between the number of solutions in $\boldsymbol{Y}_{known}$ to the number of solutions in $\boldsymbol{Y}_{true}$. Ideally, a good solution should have a value close to 1.

$$ONVGR \stackrel{\Delta}{=} \frac{ONVG}{|\, Y_{true} \,|_c} \qquad (12)$$

4) Error Ratio (*E*)

$$E \stackrel{\Delta}{=} \frac{\sum_{i=1}^{N} e_i}{ONVG} \qquad (13)$$

where N is the total number of solutions y_i found in a run, while e_i is 0 if $y_i \in \boldsymbol{Y}_{true}$ and 1 otherwise. This ratio reports the proportion of objective vectors in $\boldsymbol{Y}_{known}$ that are not members of $\boldsymbol{Y}_{true}$. Of course, a small value is preferred.

6. EXPERIMENTAL RESULTS

The two algorithms described above were implemented in a standard personal computer under a UNIX operating system using C language. For simplicity, results will be presented for only one set of data, published in [1], where it is identified as "C101." The experiments reported in this work were performed with the following parameters: $m = 10$ ant, $q_0 = 0.9$, $\beta = 1$ and $\rho = 0.1$.

To build the True Pareto-optimal Front $\boldsymbol{Y}_{true}$ both algorithms were running 200 times, as well as other tested algorithms. All found solutions were stored and from all those candidate solutions the dominated ones were erased, taking only non-dominated solutions to conform $\boldsymbol{Y}_{true}$.

To compare both algorithms, the Pareto-optimal Front $\boldsymbol{Y}_{known}$ of each algorithm was built with the 20 best runnings of each one, and the dominated solutions of each set were erased. The performance metrics were calculated for every run of each algorithm and their average values are presented in Table 1, where it is easy to note that the new approach outperforms the original MACS-VRPTW in every studied metric.

Table 1. Average metric values for the best 20 runs of 200 runnings of both algorithms: the MACS-VRPTW and the New Approach

ONVG		OTNVG		ONVGR		E	
Macs	New	Macs	New	Macs	New	Macs	New
9.75	15.85	0	1.1	0.26	0.42	1.00	0.93

Figure 2 presents three Pareto Fronts for ν = 10 vehicles, one for $\boldsymbol{Y}_{true}$ and one for each $\boldsymbol{Y}_{known}$ of both compared algorithms, considering only 20 runs. It can be seen that the $\boldsymbol{Y}_{known}$ found for the new approach is a lot closer to $\boldsymbol{Y}_{true}$ than the one calculated for the MACS-VRPTW. Even more, the whole $\boldsymbol{Y}_{true}$ is better approximated by the $\boldsymbol{Y}_{known}$ of the new approach because the $\boldsymbol{Y}_{known}$ of the MACS-VRPTW does not contain good solutions when considering F_2 and does not have enough solutions in $\boldsymbol{Y}_{true}$ or very close to it.

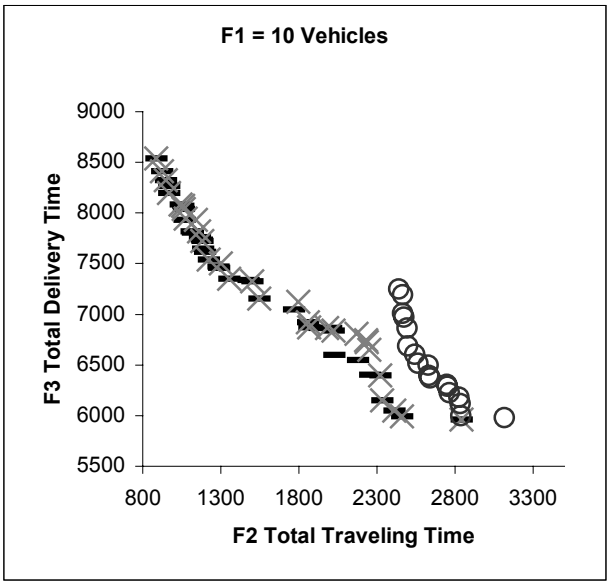

Fig. 2. Comparative Pareto Fronts. The *hyphen* represents solutions in Y_{true}; the *crosses* solutions in Y_{known} for the new approach; while the *circles* represent solutions found with the MACS-VRPTW.

Although good solutions with more than 10 vehicles were found, these solutions are not presented in Figure 2 because 2-D graphics are easier to understand, and most solutions use $v = 10$ vehicles.

Similar results as the ones presented in Table 1 and Figure 2 were obtained for other values of the parameters (m, q_0, β and ρ) and other test VRPTW problems presented in [1]. In particular, the proposed algorithm found a good number of solutions with fewer vehicles v and shorter *Total delivery time* for several test problems of [1], as *"C1-10-1"*, "*C1-10-2"*, "*C1-2-1"* and "*RC2-2-1"*.

In summary, experimental results prove that the new approach outperforms the MACS-VRPTW in the new context of a truly multiobjective problem, using the three simultaneous objective functions presented in section 2.

7. CONCLUSION

The present work modifies the MACS-VRPTW algorithm to propose, for the first time, a truly multiobjective version of the Ant Colony System for the VRPTW problem with three equally important objective functions, finding a whole set of Pareto optimal solutions, as shown in Figure 2.

Experimental results prove this new approach clearly outperforms the original MACS-VRPTW algorithm in this new multiobjective context. Moreover, the ideas presented in this work can be easily extended to other objectives and even to different multiobjective problems using the same concept of different visibilities for each objective, saving the found information of good solutions in a unique pheromone matrix. This way, it naturally combines the best solutions found so far, trying to discover new Pareto optimal solutions.

Given the promissory experimental results found so far, the authors are working on a parallel version of the proposed algorithm for a network of personal computers, using PVM (Parallel Virtual Machine), aiming to solve larger problems in reasonable time.

REFERENCES

[1] M. M. Salomon. Algorithms for Vehicle Routing and Scheduling Problems with time window constraints. Northeastern University, Boston, Massachusetts, December 1985.

[2] L. Bodin, B. Golden, A. Assad & M. Ball. Routing and Scheduling of Vehicles and Crews (1983). *The State of the Art*. Comput. Opns. Res. 10, 62-212.

[3] K. K. Lau, M. J. Kumar & N. R. Achuthan. Parallel implementation of branch and bound algorithm for solving vehicle routing problem on NOWs. *Third International Symposium on Parallel Architectures, Algorithms, and Networks*, 1997, 247-253.

[4] J. P. Pedroso. Niche search: An application in vehicle routing. *IEEE World Congress on Computational Intelligence*, 1998, 177–182.

[5] N. Yoshiike & Y. Takefuji. Vehicle routing problem using clustering algorithm by maximum neural networks. *Second International Conference on Intelligent Processing and Manufacturing of Materials,* 2, 1999, 1109–1113.

[6] L. Gomes & F.J. Von Zuben. A neuro-fuzzy approach to the capacitated vehicle routing problem. *Int. Joint Conference on Neural Networks*, 2, 2002, 1930-1935.

[7] H. Murao, K. Tohmata, M. Konishi & S. Kitamura. Pheromone based transportation scheduling system for the multi-vehicle routing problem *IEEE Int. Conference on Systems, Man, and Cybernetics*, 4, 1999, 430–434.

[8] T, Takeno, Y. Tsujimura & G. Yamazaki. A single-phase method based on evolution calculation for vehicle routing problem. *Fourth International conference on Computational Intelligence and Multimedia Applications*, 2001, 103–07.

[9] A. Gupta & R. Krishnamurti. Parallel algorithms for vehicle routing problems. *Fourth International Conference on High-Performance Computing*, 1997, 144-151.

[10] S. J. Louis, Y. Xiangying & Y.Y. Zhen. Multiple vehicle routing with time windows using genetic algorithms. *Congress on Evolutionary Computation*, 1808(3), 1999.

[11] O. Maeda, M. Nakamura, B. M. Ombuki & K. Onaga. A genetic algorithm approach to vehicle routing problem with time deadlines in geographical information systems. *International Conference on Systems, Man, and Cybernetics*, 4, 1999, 595-600.

[12] A. Chin, H. Kit & A. Lim. A new GA approach for the vehicle routing problem. *11th IEEE International Conference on Tools with Artificial Intelligence*, 1999, 307-310.

[13] K. C. Tan, T. H. Lee, K. Ou & L. H. Lee. A messy genetic algorithm for the vehicle routing problem with time window constraints. *Congress on Evolutionary Computation*, 1, 2001, 679-686.

[14] O. Arbelaitz, C. Rodriguez & I. Zamkola. Low cost parallel solutions for the VRPTW optimization problem. *Fourth International Conference on Parallel Processing Workshops*, 2001, 176-181.

[15] Z.J. Czech & P. Czarnas. Parallel simulated annealing for the vehicle routing problem with time windows. *10th Euromicro Workshop on Parallel, Distributed and Network-based Proc.*, 2002, 376-383.

[16] L. Gambardella, E. Taillard & G. Agazzi. *News ideas in optimization*. Mac Graw-Hill, London 1999, 73–76.

[17] D. A. Van Veldhuisen. Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology. Ohio, USA. May, 1999.

[18] S. Iredi, D. Merkle & M. Middendorf. Bi-Criterion Optimization with Multi Colony Ant Algorithms. *Lecture Notes in Computer Science*, Springer-Verlag, Zurich, Switzerland (March 2001) 360-370.