

Solving Multi-Objective Vehicle Routing Problem with Time Windows Using Hybrid Ants Optimization and Tabu Search Based on Performance Metrics

Mohammed Hassan Hassan¹, Laina Makdyssiian² & Wassim Habib Bilal³

¹ Professor, Department of Mathematics, Faculty of Science, Tishreen University, Lattakia, Syria

² Lecturer, Department of Communication, Faculty of Technology Engineering of Information and Communication, Tartous University, Syria

³ PhD student, Department of Mathematics, Faculty of Science, Tishreen University, Lattakia, Syria

Correspondence: Mohammed Hassan Hassan, Department of Mathematics, Faculty of Science, Tishreen University, Lattakia, Syria. E-mail: Mohdhassan11@hotmail.com, linamkdesan@gmail.com, wasem_b@hotmail.com

Received: May 7, 2017

Accepted: February 25, 2017

Online Published: April 8, 2017

doi:10.5539/mas.v11n5p52

URL: <https://doi.org/10.5539/mas.v11n5p52>

Abstract

In this research, we are studying the possibility of contribution in solving the multi-objective vehicle Routing problem with time windows, that is one of the optimization problems of the NP-hard type, This problem has attracted a lot of attention now because of its real life applications.

Moreover, we will also introduce an algorithm called Hybrid Algorithm (HA) which depends on integrates between Multiple Objective Ant Colony Optimisation (MOACO) and Tabu Search (TS) algorithm based on the Pareto optimization, and compare the presented approach is the developer with standard tests to demonstrate the applicability and efficiency.

Keywords: Ant Colony Optimization, multi objective optimization, pareto optimal set, pareto optimization, performance metrics, Tabu Search, Vehicle Routing Problem with Time Windows

1. Introduction

The Vehicle Routing Problem with Time Windows (VRPTW) is an extension of the VRP and was First introduced in 1959 (Dantzig et al., 1959).

In the VRPTW, vehicles are required to serve customers within a limited time period. Moreover it has been included in the category of NP-Hard problems. The VRPTW has many real-world applications, such as postal delivery, waste collection, school bus routing, and so on. Due to the challenging problem complexity and high practical value, the VRPTW is a very important research topic in the fields of operations research, transportation science, and computer science. (López-Ibáñez & Stützle, 2010; De Armas, Meli-Batista, Moreno-Perez & Brito, 2015; Dantzig & Ramser, 1959).

Several objectives have been considered in the VRPTW, and minimization of the number of vehicles and the total travel distance are the most common objectives in the literature. The classical way to address these two objectives is to minimize the number of vehicles first and then to minimize the total distance with the minimal number of vehicles. Optimizing the two objectives in the classical way implies that the vehicle-related cost is much higher than the distance-related cost. In many cases, however, fleet managers want to know the trade-off between these two objectives before determining the best routing plan. To accomplish this goal, another stream of research was started by searching for the set of Pareto optimal solutions rather than a single optimal solution. Hereafter, Pareto approaches refer to the approaches for which the goal is to find the Pareto set. Solutions in the Pareto set are not worse than any other in both objectives simultaneously. By looking into the trade-off between these solutions, managers can get more information and make a better decision. (Moccia, Cordeau & Laporte, 2012; Archetti & Speranza, 2014; Dantzig & Ramser, 1959).

2. The Assumptions

- 2.1 Every vehicle leaves the depot and returns to the depot.
- 2.2 The total demand of any customer is not more than the capacity of the vehicle.
- 2.3 Each customer is serviced by only one vehicle.
- 2.4 Demand of each client is q_i is known .
- 2.5 All vehicles are homogeneous with known capacities Q are used.

3. The Constraints

- 3.1 Each customer v_i is serviced only once by only one vehicle.
- 3.2 Serve every customer v_i within the time window $[e_i, l_i]$.
- 3.3 The total demand of any customer is not more than the capacity of the vehicle Q .
- 3.4 Every vehicle leaves the depot and returns to the depot v_0 .
- 3.5 The time window and capacity constraint are restricted to each vehicle .

4. The Parameters

K : is the total number of homogeneous vehicles, $K = \{k_1, k_2, \dots, k_m\}$, based in the depot v_0 .

N : is the total number of customers .

C_{ij} : is the cost to travel from node i to j .

t_{ij} : is the travel time between node i and j , ($t_{ij} > 0$), $t_{ij} = t_{ji}$.

Q : is the total capacity of vehicle.

q_i : is the demand at customer v_i , $v_i \in V$ and where $q_i \leq Q$

q_k : is the capacity of vehicle k .

e_i : is the earliest arrival time at node i .

l_i : is the latest arrival time at node i .

r_k : is the maximum route time allowed for vehicle k .

w_i : is the waiting time if vehicle k arrives early at node i .

S_i : is the service time in the node i .

In a traditional VRPTW problem, a single objective function F may be chosen from any of the following ones. (Angus & Woodward, 2009; Liang, Wang, Chen, & Tseng, 2015):

1. Number of vehicles used to serve the customers :

$$F_1 = K \quad (1)$$

where K represents the total number of vehicles.

2. Total traveling time :

$$F_2 = \sum t_{ij} \quad \forall t_{ij} \in \psi \quad (2)$$

where ψ represents a complete tour .

3. Total delivery time : (considering waiting times)

$$F_3 = \sum_{i=1}^V T_i \quad (3)$$

where T_i represents the time needed by vehicle i to return to the depot v_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 $F = (F_1, F_2, F_3)^T$ with no objective considered more important than the others.

5. Multi-Objective Optimization and Pareto-Optimal Solutions

A basic single-objective.[8,9].optimization problem can be formulated as follows :

$$\min f(x) \quad (4)$$

$$x \in S,$$

where f is a scalar function (A multi-objective problem is often solved by combining its multiple objectives into one single-objective scalar function). This approach is in general known as the weighted-sum or scalarization method, and S is the (implicit) set of constraints that can be defined as :

$$S = \{x \in R^m : h(x) = 0, g(x) \geq 0\}.$$

By having several objective functions, the notion of “optimum” changes, because in MOP, we are really trying to find good compromises (or “trade-offs”) rather than a single solution as in global optimization. The notion of “optimum” that is most commonly adopted, was originally proposed by Francis Ysidro Edgeworth in 1881. This notion was later generalized by Vilfredo Pareto (in 1896) although some authors call Edgeworth-Pareto optimum to this notion, we will use Optimum .

Multi-objective optimization can be described in mathematical terms as follows:

$$\begin{aligned} \min [f_1(x), f_2(x), \dots, f_n(x)] \quad (5) \\ x \in S, \end{aligned}$$

where $n > 1$ and S is the set of constraints defined above. The space in which the objective vector belongs is called the objective space, and the image of the feasible set under F is called the attained set. Such a set will be denoted in the following with $C = \{y \in R^n : y = f(x), x \in S\}$.

The scalar concept of “optimality” does not apply directly in the multi-objective setting. Here the notion of Pareto optimality has to be introduced. Essentially, a vector $x^* \in S$ is said to be Pareto optimal for a multi-objective problem if all other vectors $x \in S$ have a higher value for at least one of the objective functions f_i , with $i = 1, \dots, n$, or have the same value for all the objective functions. In Fig. (5.1), all the efficient solutions, is called Pareto front or Pareto curve or surface. The shape of the Pareto surface indicates the nature of the trade-off between the different objective functions. An example of a Pareto curve is reported in Fig. 5.1, where all the points between $(f_2(\hat{x}), f_1(\hat{x}))$ and $(f_2(\check{x}), f_1(\check{x}))$ define the Pareto front. These points are called non-inferior or non dominated points.

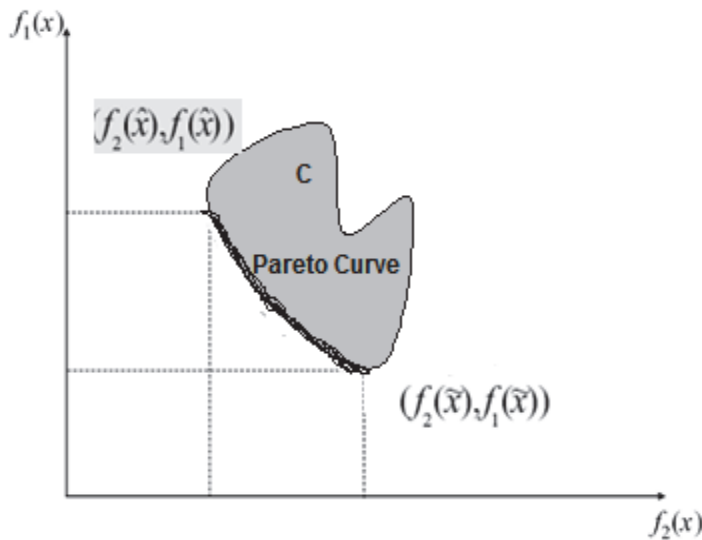


Figure 5.1. Example of a Pareto Curve

6. Classical Methods for Multiple Objective Optimization

6.1 Exact Methods

Methods to compute the Pareto set, like the Brute Force methods, exist for the majority of the optimization problems. However, in general they are only capable to manage small instances of the proposed problems, in reasonable time .[8,9].

6.2 Based Single Objective Methods

Due to the intrinsic complexity found on most of the multiple criteria problems, several techniques have been developed to take advantage from the known single objective algorithms. Those methods, where transformations from the multiple objectives problem into a single objective one are used to establish the procedures, are usually recognized as the classical techniques . Four are the most commonly classical methods to solve the

multi-optimality (Liang, Wang, Chen, & Tseng, 2015; 8,9]. All of them are based on reducing the multi-objective optimization problem to a single objective one .

6.2.1 Objective Weighting

In the former one, the multiple objective functions are combined into one overall objective function, as follows:

$$F = \sum_{i=1}^K w_i \cdot f_i(x), \sum_{i=1}^K w_i = 1, 0 \leq w_i \leq 1 \quad (6)$$

6.2.2 Distance Functions

In distance functions, the single-objective F function to be optimized is calculated using a demand-level vector, \bar{f} , which has to be specified by a decision maker:

$$F = \left[\sum_{i=0}^K |f_i(x) - \bar{f}|^r \right]^{1/r}, 1 \leq r < \infty \quad (7)$$

6.2.3 Min–Max formulation

Min–Max formulation attempts to minimize the relative deviations of the single-objective functions from the corresponding individual optimum. Hence, it tries to minimize:

$$f(x) = \max[Z_i(x)], i = 1, 2, \dots, K \quad (8)$$

where $Z_i(x)$ is calculated for a non-negative target optimal value $\bar{f}_i > 0$:

$$Z_i(x) = \frac{f_i - \bar{f}_i}{\bar{f}_i} \quad (9)$$

6.3.3 Lexicographic Approach:

The last method, Lexicographic approach (Liang, Wang, Chen, & Tseng, 2015; 9], is based on an order of importance of the objectives. Then, it attempts to optimize the i th objective keeping the best found solution for the $(i - 1)$ th objective.

7. Drawbacks Classic Methods

Since objectives have been combined to form a single objective function, a single Pareto-optimal solution can be simultaneously obtained. In real world problems, the decision-maker often requires different alternatives for the decision making, but these techniques can not offer them.

1. Even if different weights are adopted during a single run in order to obtain several solutions, the algorithm will not be able to generate concave portions of the Pareto front (Iredi, Merkle & Middendorf, 2001).
2. Furthermore, before forming the single objective from the objectives set, the decision-maker must usually have a thorough knowledge of the priority of each objective, which is a difficult problem in itself.
3. If some objectives are noisy or have a discontinuous variable space, these methods may not appropriately work.
4. If the objective functions are not deterministic, the definition of a weight vector or a demand level may become even more difficult.
5. They have much sensitivity and dependency toward weights or demand levels.

In order to solve these problems, some advanced multi-objective optimization techniques have been proposed in the last few years. They are mainly based on well-established metaheuristics such as simulated annealing, evolutionary algorithms, and ACO algorithms (Zitzler, Deb & Thiele, 2000; 9).

8. Performance Metrics

Using several measures to evaluate the performance as there is no single measure can be totally obsessed performance and efficiency to influence multiple objectives of the algorithms and there are four more common standards to evaluate multiple objectives Optimization performance.

The suit comprises the following metrics:

8.1 Overall Non-Dominated Vector Generation (ONVG)

It simply counts the number of solutions in the calculated Pareto Front, denoted as $|Y_{known}|$:

$$ONVG \triangleq |Y_{known}|_c \quad (10)$$

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

8.2 Overall True Non-dominated Vector Generation (OTNVG)

Counts the number of solutions in Y_{known} that are also in the True Pareto optimal Front denoted as Y_{true} . Because 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 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 Y_{known} .

$$OTNVG \triangleq |\{y \in Y_{known} \wedge y \in Y_{true}\}|_c \quad (11)$$

8.3 Overall Non-Dominated Vector Generation Ratio (ONVGR)

Denotes the ratio between the number of solutions in Y_{known} to the number of solutions in Y_{true} . Ideally, a good solution should have a value close to 1.

$$ONVGR \triangleq \frac{ONVG}{|Y_{true}|_c} \quad (12)$$

8.4 M_1 Metric

M_1 metric represents the distance between the result of an algorithm, denoted as Y , and the Pareto front (\bar{Y}). This metric is based on Eq(8.13), in which $|Y|$ means the number of non-dominated solutions in front of Y . The smaller M_1 metric is, the smaller difference between \bar{Y} and Y is.

$$M_1(Y) = \frac{1}{|Y|} \sum_{p \in Y} \min\{||p - \bar{p}|| | \bar{p} \in \bar{Y}\} \quad (13)$$

8.5 M_2 Metric

M_2 metric evaluates the distribution of solutions in the PF returned by an algorithm (denoted as Y). This metric is based on Eq (8.14), in which parameter σ is a positive constant. The larger M_2 metric is, the wider the coverage of the obtained solutions is.

$$M_2(Y) = \frac{1}{|Y-1|} \sum_{p \in Y} |\{q \in Y; ||p - q|| > \sigma\}| \quad (14)$$

8.6 M_3 Metric

M_3 metric is used to evaluate the diameter of PF returned by an algorithm (denoted as Y) based on Eq (8.15), in which p_i denotes the solution value in p for objective i . The larger M_3 metric is, the larger region of the objective space of solutions locate.

$$M_3(Y) = \sqrt{\sum_{i=1}^b \max\{||d(p_i, q_i)||; p, q \in Y\}} \quad (15)$$

8.7 C Metric

C metric is devoted to compare the performance of two algorithms by calculating the dominance degree of their respective PF. In Eq (8.16), Y_1 and Y_2 represent PFs returned by two different algorithms.

$$C(Y_1, Y_2) = \frac{|\{p_2 \in Y_2; \exists p_1 \in Y_1; p_1 < p_2\}|}{|Y_2|} \quad (16)$$

8.8 Error Ratio (ER)

ER measures knowledge as follows:

$$E \triangleq \frac{\sum_{i=1}^N e_i}{ONVG} \quad (17)$$

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

Note that the decline in the value of error rate represents the best collections no-dominated.

Shows Table (1):Average metric values for the Multi-objective Ant Colony optimisation MOACO and Pareto optimal for ant colony algorithm PO-ACO and arranged table from best to worst for each metric as shown

Table 1. Benchmark algorithms known. (Pinto & Barán, 2005). During the 60 iterate .

Benchmark algorithms	M1	M3	M2	M3	Error
MAS	0,1603	0,6273	0,4873	0,6273	0,9977
POACO	0,2139	0,4081	0,2889	0,4081	1.000

9. Processing the Problem

After the researchers studied the different algorithms separately, they researchers realized that it is not enough, and the researchers believe that the hybridization of this Heuristic algorithms and metaheuristic can lead to better results through the integration of these algorithms and the combination of their different function (Angus & Woodward, 2009; Tan, Chew & Lee, 2006; Bouhafs, Hajjam & Koukam, 2010; Baran & Schaerer, (2003) And we will propose in this paper a hybrid algorithm developed from Heuristic algorithms and metaheuristic to resolve the Multi-Objective Vehicle Routing Problem With Time windows, The proposed algorithm will form a different framework of processing .

Before starting our algorithm we'll provide some basic approximate algorithms.

9.1 Tabu Search TS

Tabu Search (TS) is a local search metaheuristic introduced by Glover (1986) to guide the local search method to continue its search beyond a local optimum. (Moccia, Cordeau & Laporte, 2012). TS is a memory-based search strategy ,where we start from an initial solution replaces the dissolution of an adjacent new if its the best, which usually leads to find a local optimal solution , and form the list of search outlawed memory structure, using short-term memory to prevent cycling is used in many other meta-heuristic approaches as well.

a set of rules and short-term solutions that has had in the recent past and the campus of use: The memory structure used in the Tabu search can be divided into three categories:

1. Short term memory structure: Is a list of solutions that recently examined, and if there is a possible solution in this list, it can not be reconsidered until it reaches the completion point.
2. Memory medium-term structure: Is a list of the foundation aims to bias towards research in the most promising areas of the search space.
3. Long-term memory structure is a list of foundations that promote diversity in the search process.

Memory is short-term enough to achieve an advanced solution for a solution created by local traditional methods, but the medium and long-term memory structure are often necessary to resolve difficult problems,

with medium brown-term work in the first place to intensify and diversify the search, and we will focus in this paper on short-term memory to diversify the search and avoid falling into local optimization, and will probably include long-term memory structure to focus clearly and diversification as is the case in some advanced search algorithms outlawed.

9.2 Pseudocode Tabu Search (TS). (Moccia, Cordeau & Laporte, 2012)

- 1- *Initial = Initialsolution().//Initial is current solution*
- 2- *S = Localsearch(Initial). //Make sure We start with a local minimum*

```

3-   $S^* = S // S^*$  represent the best solution found
4-  While not StoppingCondition()do
5-    moves = RankMoves(s);
6-    Moved=false ;
7-    While not Moved do
8-      m = head(moves);
9-      if IsNotTabu(m) Then
10-     Moved=true ;
11-     InsertTabuList(m);
12-     If  $O(s) < O(S^*)$  then  $S^* = S$ 
13-     Return  $S^*$ 
14-   end

```

10. Moaco Generalization

Most of the applications of real-life have problems with multiple conflicting objectives, and will be considering the vehicle Routing problem with time windows along with a multi-objective stretch. This work draws inspiration from the approach proposed by Baran and Schaerer in 2003 in order to solve the biobjective TSP (Pinto & Barán, 2005; Maniezzo, Gambardella & De Luigi, 2004; Iredi, Merkle & Middendorf, 2001; Baran & Schaerer, (2003), which consists in a MOACO algorithm with a single ant colony, a single pheromone structure τ and different adaptive visibility η for each objective function.

And each one represents one objective in order to reduce the cost of each continued objective with equal importance so a whole Pareto-optimal solution set can be found in only one run of this meta-heuristic. Every objective share the same pheromone trace, represented in a single matrix $\tau = \tau_{ij}$.

10.1 Ant Colony Algorithm for Multiple Objectives: (Zitzler & Thiele, 1999)

Pseudocode Multiobjective ACO

1. Set parameters;
2. Pareto_set = empty.
3. Initialize_ant .
4. $\tau_{ij} = \tau_0$, / Initialize pheromone .
5. Initialize heuristic matrix η ;
6. while termination criteria not met do
7. ConstructAntSolution();
8. ApplyLocalSearch() (optional);
9. UpdateParetoSet();
- 10 . UpdateGlobalPheromone();
11. Return the Pareto set P

11. Construct a Solution

To build possible solutions using a variety of vehicles According to the Constraints ,at each generation (or iteration) of the proposed algorithm, every ant h (of a set of H ants) constructs a feasible solution, beginning its tour in a Depot (d_0) and successively choosing a following node (or Depot to return) v_j , out of the set of feasible nodes not yet visited N_i^h where the sub index i indicates that the ant h is in the node v_i . At each node v_i , the set N_i^h is calculated in order to discover every node (or client) to which merchandise still needs to be taken and are not in violation of any restriction . The set N_i^h does not include the Depot until a tour has been finished . Naturally, when a vehicle cannot visit any more nodes, it must return to the Depot and therefore, a solution ψ has been found .An ant h chooses to move from one node v_i to another node v_j using both: the heuristic information and the pheromone traces. The heuristic information is given by the visibility η_{ij} , while the pheromone information is given by τ_{ij} , to choose the next node v_j to be visited by an ant h which is found in v_i , each node or branch is assigned a probability p_{ij} giving in (16) , and in this manner, the next branch v_j

is chosen from N_i^h at random using that probability. (Liang, Wang, Chen, & Tseng, 2015; Pinto & Barán, 2005; Maniezzo, Gambardella & De Luigi, 2004) originally given for only 2 objectives must be generalized here for a number u of objective functions, the probability p_{ij} can be generalized to u objectives in the following manner:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha [(\eta_{1ij})^{\lambda_1} (\eta_{2ij})^{\lambda_2} \dots (\eta_{uij})^{\lambda_u}]^\beta}{\sum_{g \in \mathcal{N}_i^k} (\eta_{1ig})^{\lambda_1} (\eta_{2ig})^{\lambda_2} \dots (\eta_{uig})^{\lambda_u}} & \text{if } j \in \mathcal{N}_i^k \text{ (18)} \\ 0 & \text{otherwise} \end{cases}$$

where

u is the number of objective functions of the problem, while η_{uij} is the visibility related to the objective function u for the link (i, j) . the variables $\lambda_1, \lambda_2, \dots, \lambda_u$ define the relative influence among visibilities. parameter α defines the relative influence of the pheromone traces.

When generalizing Baran and Schaerer’s original proposal. (Maniezzo, Gambardella & De Luigi, 2004; Iredi, Merkle & Middendorf, 2001). for u objectives ($u \geq 2$), each ant utilizes one of the $(\sigma + 1)^u$ different combinations for the visibility, $\lambda \in \{0, 1, \dots, \sigma\}$ in order to improve the exploration of the search space. (Maniezzo, Gambardella & De Luigi, 2004; Iredi, Merkle & Middendorf, 2001; Bouhafs, Hajjam & Koukam, 2010). Consequently, it is

recommended to use $H = (\sigma + 1)^u$ ants in order to cover all possible variants found in Table I.

The value taken by each variable of relative influence λ varies with the number h of the ant building a solution, achieving in this way the desired characteristic of having each ant h having preferences on different areas of the exploration domain. This way, each ant h potentially aims at different regions of the Pareto front that is being searched.

In consequence, the proposed solution considers generations in which H ants construct solutions, each with a different probability distribution, expecting in this way to improve the exploration of the MOACO.

solutions in the best known Pareto set approximation to verify if it is a no-dominated solution. If, in effect, it is a new optimal Pareto solution, it is included in a set P and dominated solutions are erased from the current set P of optimal solutions.

11.1 Pheromone Updating

After each generation of the proposed algorithm is finalized. (Pinto & Barán, 2005; Maniezzo, Gambardella & De Luigi, 2004) an update of the pheromone matrix is carried on with the no-dominated solutions contained in the Pareto_Set, according to the equations:

$$\tau_{ij}^* = \tau_{ij} + \Delta\tau_{ij} \tag{19}$$

$$\tau_{ij}^{**} = (1 - \rho)\tau_{ij}^* \tag{20}$$

For this calculation, the objective functions are normalized between a very small value ε and 1; this is, to the interval $[\varepsilon, 1]$, for instance, utilizing the following equation:

$$F_i(\psi) = 1 - (1 - \varepsilon) \frac{F_{iMAX} - F_i(\psi)}{F_{iMAX} - F_{iMIN}} \tag{21}$$

Therefore, to calculate the value of $\Delta\tau_{ij}$ the following equation may be used:

$$\Delta\tau_{ij} = \frac{1}{\sum w_i F_i(\psi)} \tag{22}$$

where w_i is a weight corresponding to the relative importance that can be attributed to each objective function $F_i(\psi)$ and that can be defined at the discretion of the decision maker or at random each time it is used. (Iredi, Merkle & Middendorf, 2001; Angus & Woodward, 2009).

12. The General Procedure for Finding No-Dominated Solutions

General procedure has been put to find a No-Dominated Solutions as follows:

Pseudocode Find no-dominated solutions. (Maniezzo, Gambardella & De Luigi, 2004; Tan, Chew & Lee, 2006).:

1. From $k = 1$ m.
2. From $k' = 1$ m.
3. If $f_1(\psi_{w,k}) \leq f_1(\psi_{w,k'})$ &

$$f_2(\Psi_{w,k}) \leq f_2(\Psi_{w,k'}) \& \dots f_n(\Psi_{w,k}) \leq f_n(\Psi_{w,k'})$$

Then

4. if $f_1(\Psi_{w,k}) < f_1(\Psi_{w,k'})$ &
 $f_2(\Psi_{w,k}) < f_2(\Psi_{w,k'})$ & ... $f_n(\Psi_{w,k}) < f_n(\Psi_{w,k'})$ Then
 Mark solution $\Psi_{k,k'}$ as dominated
5. Update_Tabu_List.
6. End If
7. End If .
8. End From ' .
9. End From .
10. Save set of solutions not marked as Undominated_set. Return Set of no-dominated solutions (Pareto set approximation).
11. End.

13. Proposed Algorithm

We offer a development of a hybrid algorithm (ACO-TS) –PO based on the principle of complementarity between Multi-objective Ant Colony optimisation MOACO and tabu search (TS) algorithm based on the Pareto optimization PO , and Used performance evaluation standards(criteria).

And it characterized the proposed algorithm for the basic algorithm in the following points:

1. Considering all the objectives with the same importance.
2. Finding a set of Pareto optimal solutions P may be found.
3. using an single ant colony to minimize all three functions (the multiple objectives)simultaneously
4. All objectives share the same pheromone trails and three heuristic matrices, one for each objective.

The main difference of the basic algorithm with proposed is to consider all the objectives with the same importance, no objective has any precedence over the others, and to Find a set of Pareto optimal solutions P may be found, the proposed new algorithm uses colony of ants alone to minimize all the consequences of the objective at the same time and all objectives share the same pheromone trails, knowing the good solutions is important for each objective In front Pareto.

Multi-objective Ant Colony optimisation MOACO offers solutions of high quality, and therefore dependent as a component of a hybrid model, On the other hand tabu search algorithm TS includes memory able to adapt and responsive to the exploration and the adaptive memory allows the tabu search algorithm TS to reduce the Iterations(cycles) of the solution, And able to perform Radical(Radical) changes on the basis of previous information, and responsive exploration allows, tabu search algorithm TS to apply diversification and concentration strategies, And we will consider Multi-objective Ant Colony optimisation algorithm MOACO and tabu search algorithm TS to form a hybrid mix different and integrated in nature.

13.1 The Proposed (ACO-TS)-PO Algorithm

The pseudocode for the proposed algorithm (Multi-Objective Ant Colony Optimization for the Vehicle Routing Problem with Time Windows) **ACO-TS** -VRPTW follows :

13.2 Hybridization Steps Proposed Algorithm Developed: (ACO-TS)-PO

Pseudocode HMOACO-TS-VRPTW

1. Read entry parameters
2. Pareto_set = empty.
3. Tabu_List= empty //Initialize_Tabu_List .
4. Initialize_ant .
5. $\tau_{ij} = \tau_0$,/ Initialize pheromone matrix , $\forall (i,j) \in E: G(V,E)$
6. From $c = 1$ until n // The number of iterations
7. From $k = 1$ until m // Number of ants

8. Construct solution $\psi_{c,k}$.
9. Update_Tabu_List (New_Solution)
10. Calculate objective function $F(\psi_{c,k})$.
11. End From k .
12. Find no-dominated solutions (from the Quantity_of_Ants available) .
13. Update Pareto_Set .
14. Update pheromone matrix τ_{ij} .
15. Update_Tabu_List .
16. End From c .
17. Save Pareto_set of solutions .
18. Exit .
19. Return Pareto set .

// Pareto set of solutions $\psi(\mu)$ with their respective objective functions $F(\psi(\mu)) = [F_1(\psi(\mu)), F_2(\psi(\mu)), \dots, F_u(\psi(\mu))]$,

$\mu \in N - \{0\}$ represents the number of solutions within the *Pretoset*.

14. Experimentation and Results

These experimental results have been conducted, using C++ language, on a PC using a wizard corei3 and 2 GB of RAM, and some of the inputs and outputs Benchmark instances (BEST KNOWN SOLOMON SOLUTION; Solomon). The vehicle capacity, locations and time-windows of the customers and depot are those specified in the Solomon instances. Based on the calculation results of 56 record the event for the researcher Solomon's, to eight different types are RC201, RC202, RC203, RC204, RC205, RC206, RC207, RC208, and have been selected from a hundred state geographically dispersed, and has compared our results with the known record results. (Solomon)_ (BEST KNOWN SOLOMON SOLUTION). To evaluate the performance of the proposed algorithm in solving the Multi-Objective Vehicle Routing Problem With Time windows, the gap has been calculated according to the relationship:

$$Gap = \frac{\text{Proposed Algorithm} - \text{bestknown Benchmark results}}{\text{Proposed Algorithm}} * 100$$

Note that the proposed algorithm has been setting to stop at 60 to iterate on the Table(2).

Table 2. The experimental results of the proposed algorithm is compared with the best known Benchmark results and the calculation of the gap between them

Type	The Best Known Benchmark Results	The Proposed Algorithm	Gap
RC201	54905	58244	5.7327793
RC202	53404	57877	7.7284586
RC203	53620	55453	3.305502
RC204	54778	58698	6.6782514
RC205	51907	57728	10.083495
RC206	50507	56798	11.076094
RC207	51453	52596	2.1731691
RC208	52501	55733	5.7990777

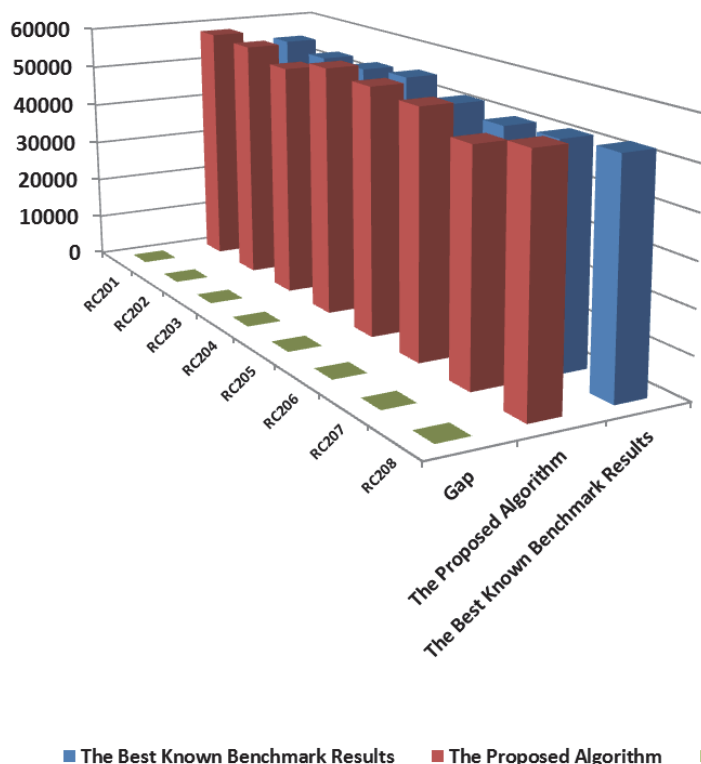


Figure 3. Scheme to compare the proposed algorithm and with the best known Benchmark results

Table 3. Application performance evaluation standards and the error rate on the proposed hybrid algorithm

The Proposed Algorithm	M1	M2	M3	Error
MOACO-TS	0,1354	0,5583	0,7418	0,9534

And the experimental results shown in the table (2) and (3) that the proposed algorithm is better and more efficient and effective, especially to the Vehicle Routing Problem with Time Windows with multiple objectives of Large measurement. And give the best results within a reasonable time and offers the possibility to achieve the best in terms of the speed of convergence performance, And the ability to find the best solutions Of algorithms that were compared with , and there was a small gap between this proposed algorithm and The Best Known Benchmark Results, excelled in this work in the average distance to the front Pareto, The solutions that have been obtained for the Vehicle Routing Problem with Time Windows with multiple objectives got the lowest average distance of fronts Pareto, beating The Best Known Benchmark Results as shown in Table1, and the metric measure M_1 proves that the proposed hybrid algorithm produced an average of fronts Pareto closer to the optimal level of other algorithms, and that the low value of the error rate represents the best sets no-dominated .

15. Conclusion

We have presented modifies the vehicle Routing problem with time windows along with a multi-objective stretch , 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 .

Experimental results prove this proposed Hybrid algorithm clearly outperforms the original MACO-VRPTW algorithm in this new multiobjective context. Moreover, the ideas presented in this research 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, combines naturally the best solutions found so far, trying to discover new Pareto optimal solutions.

The proposed algorithm has more efficient results compared with the known benchmark problems results . It also gives better results, within a reasonable time. The results showed that the efficiency of the proposed algorithm is the best where it increased performance and reduced cost .

References

- Angus, D., & Woodward, C. (2009). Multiple objective ant colony optimisation. *Swarm Intelligence*, 3(1), 79–83.
- Archetti, C., & Speranza, G. (2014). A survey on matheuristics for routing problem. *EURO Journal on Computational Optimization*, 2, 235–246.
- Baran, B., & Schaerer, M. (2003). A multiobjective ant colony system for vehicle routing problem with time windows. In *Proceedings of the Twenty-first IASTED International Conference on Applied Informatics*, Austria, 97–102.
- BEST KNOWN SOLOMON SOLUTION. Retrieved from <http://www.sintef.no/Projectweb/TOP/Problems/VRPTW/Solomon-benchmark/100-customers/>
- Bouhafaf, L., & Hajjam, A., & Koukam, A. (2010). A Hybrid Heuristic Approach to Solve the Capacitated Vehicle Routing Problem. *Journal of Artificial Intelligence: Theory and Application*, 1, 31-34.
- Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6(1), 79-89.
- De Armas, J., Meli-Batista, B., Moreno-Perez, J. A., & Brito, J. (2015). Gvns for a real-world rich vehicle routing problem with time windows. *Engineering Applications of Artificial Intelligence*, (42), 45-56.
- Doerner, K., Gutjahr, W. J., Hartl, R. F., & Strauss, C. (2004). Stummer, C. Pareto ant colony optimization. A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*, (131), 79–99.
- Iredi, S., Merkle, D., & Middendorf, M. (2001). Bi-Criterion Optimization with Multi Colony Ant Algorithms, *First International Conference on Evolutionary Multi-criterion Optimization (EMO'01)*, 1993, Lecture Notes in Computer Science, 359-372.
- Liang, R. H., Wang, J. C., Chen, Y. T., & Tseng, W. T. (2015). An enhanced firefly algorithm to multi-objective optimal active/reactive power dispatch with uncertainties consideration. *International Journal of Electrical Power & Energy Systems*, (64), 1088–1097.
- López-Ibáñez, M., & Stützle, T. (2010). Automatic configuration of multi- objective ant colony optimization algorithms, pp. 95–106.
- Maniezzo, V., Gambardella, L., & De Luigi, F. (2004). Ant Colony Optimization”. En: Onwubolu, G. C., & B. V. Babu(Eds.): *New Optimization Techniques in Engineering*, Springer-Verlag, Berlin Heidelberg, 101-117.
- Moccia, L., Cordeau, F., & Laporte, G. (2012). An Incremental Tabu Search Heuristic for the Generalized Vehicle Routing Problem with Time Windows. *Journal of the Operational Research Society*, 238-244.
- Pinto, D., & Barán, B. (2005). Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach”. LANC'05, Cali, Colombia.
- Solomon, M. M. Solomon instances described. Retrieved from <http://web.cba.neu.edu/~msolomon/problems.htm>
- Tan, K. C., Chew, Y. H., & Lee, L. H. (2006). A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *ComputOptim. Appl.*, 34, 115–151.
- Zitzler, E., & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach”, *IEEE Trans. Evolutionary Computation*, 3(4), 257-271.
- Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2), 173–195.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).