Novel Solutions for Capacitated Vehicle Routing Problem using an Ant Colony Optimization Algorithm

[1]Ashima Gupta, [2]Sanjay Saini
[1]Research Scholar, [2]Assistant Professor
[1]Department of Physics & Computer Science,
Dayalbagh Educational Institute, Agra, 282005, India

Abstract. This article presents an enhanced ant colony optimization (eACO) algorithm for solving the capacitated vehicle routing problem (CVRP). CVRP is the core component of VRP, and also a difficult combinatorial optimization problem. An enhanced ACO algorithm is implemented on five CVRP benchmark problems, improving several of the best-so-far results existing in the literature. The computational results show that the enhanced heuristic can produce optimal solutions when compared to other existing heuristics. Results indicate that the proposed heuristic is an alternative to solve CVRP.

Keywords: Capacitated vehicle routing problem, CVRP, heuristics, enhanced ACO, optimization, optimal solution.

I. Introduction

The Vehicle Routing Problem is a widely examined problem in the field of operation research and combinatorial optimization. VRP is a class of problems including the plan of optimal routes for a fleet of vehicles to benefit a set of customers subject to side constraints. It has significance in various important domains, e.g. transportation, travelling, distribution and logistics [1] etc.

Reflecting the huge variety of conditions, a large number of extensions of VRP exist, depending on the nature of the delivered goods, required service quality and the type of customers and vehicles. Some typical complications are: deliveries within a specific time window (VRPTW) [2], customers require pickups and deliveries (VRPPD), backhauling (VRPB), vehicles sitting at multiple depots (MDVRP), split delivery (SDVRP) [3] etc. In all these cases, the goal is to provide services at minimum cost. In the vast landscape of variants, capacitated VRP occupies a central position. Therefore, the complexity as well as the importance of this variant has motivated many people, to find all possible methods to solve the capacitated vehicle routing problem optimally.

Some of the proposed exact and approximate methods that can lead to optimal solutions to the CVRP include: branch-and-bound (B&B), branch-and-cut (B&C) [4], branch-and-price (B&P) [3, 5, 6]. However, these exact algorithms can solve only small size instances optimally and not suitable for large size instances having very high computational complexities of VRPs [1, 2, 7]. Therefore, researchers switched to exceptionally efficient nature inspired techniques based on the intelligence present in ants, birds, bacteria, bees, water drop etc. These strategies are capable to give optimal solutions for large and complex problems in tractable time window.

In the last two-three decades an increasing number of meta-heuristics have been developed to solve the CVRP. The work can be categorized into simulated annealing [8], tabu search [1, 9, 10], large neighborhood search [11], variable neighborhood search [12], genetic algorithm [13, 14], evolutionary algorithms [15, 16, 17], particle swarm optimization [18, 19, 20] ant colony optimization [21, 22, 23, 24], artificial bee colony [25] etc. VRP meta-heuristics broad overviews can be seen in various survey papers [3, 26, 27]. Table 1 highlights some well-known ACO based algorithms for CVRP.

Table 1. Some well-known ACO algorithms for CVRPs [26, 48]

Author	Year	Method	Results/Comparison
Bullnheimer	1999	Ant System (AS)	A competitive solution approach for CVRPs
Bell et. al.	2004	Multi Colony ACO	Better approach for large problems
Reimann et.al.	2004	Savings Based AS	Solved large and complex VRPs optimally
Manfrin	2004	ACO	Results found better than 5 heuristics
Doerner et.al	2006	Parallel ACO	Parallel solutions better than serial solutions
Yu,Yang, Yao	2009	Improved ACO (IACO)	Efficient hybrid approach for VRPs
Bin et.al.	2009	Ant_weight + GA	Improved solutions by exploring search space
Zhang & Tang	2009	SS_ACO + NS	Competitive to produce quality solutions
Bouhafs	2010	ACO + Savings + LS	Improvement in results by local search
Ren et. al.	2010	ACO + Local search	Results are better than other heuristics
Kanthavel	2011	Nested_PSO	Proved as better meta-heuristic
W. F. Tan	2012	ACO + Swap + 3-opt	Found quality solutions in reasonable time
Gomez & Salhi	2014	New_ABC	Better than original ABC & other heuristics
Wang et. al.	2016	AMR + Savings (SA)	Much Efficient than existing algorithms
Teymourian et.al.	2016	IWD + Cuckoo + LSHA	Got 90% optimal solutions on benchmark instances
Gupta & Saini	2017	ACO + 2-Opt + Swap + Memory + Trail reset	Efficient algorithm for optimal solutions also improved existing best known solutions

Designing a superior approach for the CVRP may decrease the cost of goods, travel, transportation, making striking effects on our economy. In the pursuit to accomplish better optimal solutions, the present endeavor is aimed to tackle the CVRP using an improved ACO algorithm. The performance of the algorithm is evaluated on five different benchmark instances, proposed by Augerat in 1995 (set A, set B) [28], Christofides and Eilon in 1969 (set E) [29], Christofides, Mingozzi and Toth in 1979 [30] and Fisher in 1994 (set F) [31] and the results are compared with the results of other heuristics available in literature.

The article is organized as follows: Section 2 gives the CVRP formulation and highlights the objectives and constraints connected to the problem. In Section 3 the refinements made to basic algorithm to make it enhanced ACO algorithm is given. Definition of variables and improved results with their comparisons are presented in Section 4 and the last section gives the conclusion and future work scope.

II. CVRP Formulation

CVRP is the most elementary variant of VRP in which the fleet of vehicles have same capacity limits. Formally, the CVRP can be defined as [1]**:**

Graph: the problem is defined on an undirected graph $G = (V, E)$,$V = \{v_0, v_1, \ldots, v_n\}$ is the set of vertices and E is an edge set.

Depot: in the graph, vertex v_0 stands for the depot from where a route begins and ends.

Clients: the problem is characterized for $n$customers presented by vertices $v_1, v_2 \ldots v_n$. Every client has a non-negative deterministic mand q.

Vehicles: each vehicle has maximum capacity Q. Vehicles can serve many clients, however, the total of demands to every client should not surpass the vehicle capacity Q.Also, the vehicle must begin and end at the depot.

Travelling Cost: C_{ij} represents the cost of travelling between customers i and j. It is mostly figured out using Euclidian distance between the clients.

Route: starting from the depot, constructed of sequence of visited nodes and finally ending at the depot. The length of each route r relies on the number of clients.

CVRP: each vehicle has restricted capacity. It guarantees that the sum of customers' requests q_i can't surpass the vehicle capacity Q. Also, the aggregate route distance d_{ij} of a vehicle can not exceed its route length constraint. It likewise guarantees that every client can be served by just a single vehicle.

The CVRP is solved to accomplish number of objectives while considering certain constraints that are expressed below:

- *Objectives:*
  - Minimize the total cost of travelling.
  - Minimize the total number of vehicles.
  - Minimize the distance travelled by all vehicles.
- *Constraints:*
  - Every client should be visited only once.
  - Each vehicle must begin and end at the depot.
  - Total requests of clients of any route don't exceed the vehicle capacity.

III. Enhanced ACO Algorithm

Marco Dorigo proposed the ACO algorithm in 1992 [32], which aims to find optimal solutions in a graph, based on the conduct of ants looking for a path between their colony and a food source [2].The colony mates communicate to each other with the help of a trace known as pheromone. Pseudo-code for the original ACO algorithm is given below:

```
Procedure ACO_Algorithm
    while(not_termination)
        constructSolution()
        applyLocalSearch()
        pheromoneUpdate()
    end while
end procedure
```

In the proposed method, solutions are enhanced by considering several factors:

1) Two cities (customers) from different routes are exchanged using 1-1swap heuristic, i.e. c_1 (city) from t_1 (tour) is swapped with c_2 from t_2, if it can improve the solution.

2) After few iterations ants won't explore some edges because of lower pheromones, consequently can stuck in local minima. Therefore, in avoidance to being trapped in local minima, pheromone will be reset (based on Bullnheimer ACO algorithm) for all the edges [21] and to achieve exploitation, pheromone increased for the edges that found best solution so far, by some factor.

3) Each ant has an associated memory to record the current solution (which can be further improved) and a count variable (no. of iterations for which solution is not improved). Hence, solution is improved in each iteration rather than building a new solution.

The pseudo code for enhanced ACO algorithm is given as:

```
1)  Initialize Parameters
        For maximum Iterations:
2)  Solution Construction:
        For each ant:
            if Previous_Solution = Null
                Build New Solution starting from depot
            else Improve Previous_Solution as follows:
            a) Choose new edge not in Previous_Solution that lead to maximum saving
            b) New_Tour = Old_Tour + new edge + Build remaining solution
3)  Apply Local Search: 2-opt + Swap
4)  Update Memory
     For each ant:
        Previous_Solution = New_Solution
        if  New_Solution_Cost < Previous_Solution_Cost
            Count = 0            % no. of times solution not improved
        else Count = Count + 1
            if Count >Max_Count
```

Previous_Solution = Null
Count = 0; % reset count

5) *Update Pheromone*
6) *Reset Pheromone*
if Iteration % Max_R ==0
Reinforce Pheromone for each edge (i , j) as:
if (i, j) belongs to best Solution
Pheromone (i , j) = Initial + New Pheromone
else
Pheromone (i , j) = Initial Pheromone

After initializing the enhanced ACO algorithm, two basic steps: (i) route construction and (ii) pheromone update, are repeated for the given set of iterations. For initial placement, the number of artificial ants kept equal to the number of customers, so that one ant can be placed at one customer, at the start of the iterations. To improve the performance and to reduce the computational time of the algorithm a 2-opt local search is included.

Each ant starts at some random vertex v and then selects one edge from its neighborhood using probability p,given as:

$$p_{i,j}= \begin{cases} \dfrac{[T_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k\epsilon\Omega}[T_{ik}]^{\alpha}[\eta_{ik}]^{\beta}} & if\ v_j \epsilon\ \Omega \\ 0 & otherwise \end{cases}$$

Here, $p_{i,j}$ is probability for choosing an *edge* (i,j), which is biased by α and β parameters, that determine the relative impact of the trails and the visibility respectively. T_{ij} is pheromone trails deposited on edge (i, j) and $\eta_{ij}$is the visibility of *edge* (i,j), which is defined as reciprocal of distance[22]. Parameters f and g are used for the visibility as:

$$\eta_{ij} = d_{ik} + d_{kj} - g * d_{ij} + f * | d_{ik} - d_{kj} |$$

Ants choose next cities, until each city has been visited. Whenever, the choice of city leads to infeasible solution due to capacity or total length constraint, a new tour is created.

The pheromones are updated by elitist ants only, ranked according to solution quality. The updating rule is given as:

$$\mathrm{T}_{\mathrm{ij}}^{\mathrm{new}} = \rho \mathrm{T}_{\mathrm{ij}}^{\mathrm{old}} + \mathrm{a} \sum_{\mu=1}^{\sigma-1} \Delta\mathrm{T}_{\mathrm{ij}}^{\mu} + \mathrm{b}\,(\sigma * \Delta\mathrm{T}_{\mathrm{ij}}^{*})$$

$$\Delta\mathrm{T}_{\mathrm{ij}}^{\mu} = \frac{(\sigma - \mu)}{L_{\mu}}$$

Here, ρ is the trail persistence $(0 \leq \rho \leq 1)$, thus the pheromone evaporation can be calculated as $(1-\rho)$. Δ_{ij}^{μ} is the amount by which pheromone increased on an *edge* (i,j) visited by μ^{th} best ant and L_{μ} is the best solution found by the best ant.

$\Delta\mathrm{T}_{\mathrm{ij}}^{*} = {}^{1}/_{L^{*}}$ is the amount by which elitist ants increases the pheromone, if an *edge* (i,j)belongs to the best-so-far solution. $L^{*}$is the objective value of best-so-far solution found. Here, a and b are the scaling factors.

IV. Solving CVRP with Enhanced ACO algorithm

In this section the benchmarks taken under study, the parameters setting applied for the enhanced ACO algorithm and the results obtained and their comparison with other existing heuristics is discussed.

4.1 Benchmark Problems

The enhanced ACO algorithm was tested on standard benchmarks of CVRP. These include five Euclidean distance type VRP instances described in Augerat *set A* and *set B* [28], Christofides and Eilon *set E* [29], Christofides, Mingozzi and Toth (*CMT)*[30] and Fisher *set F* [31]. All the considered instances are freely available at CVRP library created by Ivan Xavier [33].

Set A has 27 different instances with number of customers ranging from 32 to 80, having general type demands and Euclidian distances and set B contains 23 instances with maximum of 78 customers and a depot. From set E, 11 instances were tested in which number of customers ranging from 22 to 101. Set F has 3 instances with 44, 71 and 134 customers.

The last set consist of 14 different problems contain 50 to 199 customers and an additional service point. Customers are randomly distributed in the plane for first 10 problems, but they are clustered in other 4 problems. Problems 1-10 are identical, except that problems 6-10 have route length constraint i.e. route of each vehicle is bounded, while the former problems are free from this restriction. On the other hand, the clustered problems 13 and 14 are the counterparts of 11 and 12, with tour length constraint.

In the following figures of CVRP solutions, the depot is pointed by a bold square and customers around the cities are marked by a circle. The straight line connects the route traversed by vehicles from one customer to another. X-axis and Y-axis (figure 2 onwards) show the x and y coordinates of customers' respectively.

4.2 Parameters Used

Enhanced ACO heuristic has been coded in MATLAB 2015 and experiments were performed on 2.93 GHz i7 octa-core computer. In this, *M* artificial ants are used, which are initially placed at customers $v_1, v_2 \ldots v_n$. The candidate list size i.e. nearest neighborhood of each city was set to *N/4*, i.e. only one fourth locations (the closest ones) were considered.

The initial pheromone concentration is tuned to $T_0 = 1.0$ (started with *0.92*), as it is a decent practice suggested by Dorigo et. al. [22] to set the initial pheromone to a value that is slightly higher than the expected measure of pheromone deposited i.e. $\rho = 0.9$ (tuned, started with

0.80) by the ants in one run. Further, to achieve exploitation the pheromone is raised by $T_1 = 1.2$ (tuned from *2.0*) for the edges belonging to best-so-far solution.

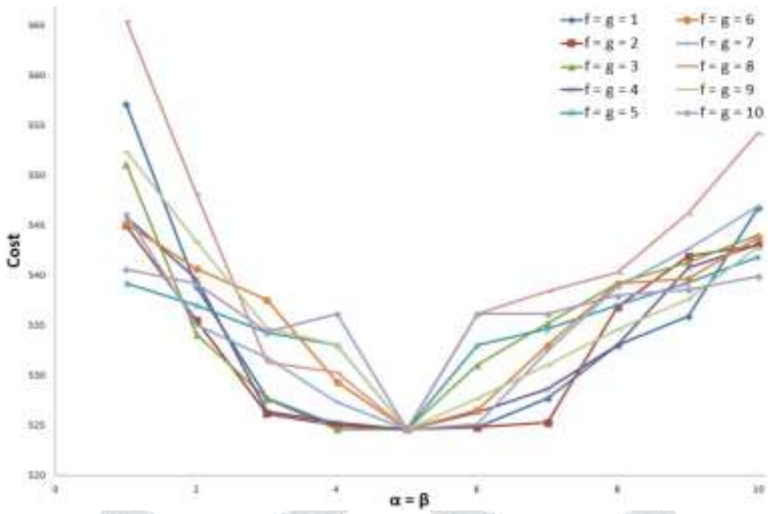

Figure 1. Shows the plot for Cost when f = g are fixed and $\alpha = \beta$ varies from 1 to 10.

To reduce the number of parameters to tune, α is set equal to β $(\alpha = \beta)$. The performance of the heuristic is then evaluated using different values of α and β in the interval from 1 to10. It is found that the performance of the heuristics is insensitive over this range for many test instances, and for the majority of the test instances setting $\alpha = \beta$ to *5* seems to be the best choice. Similarly, for parameters *f* and *g*, we tested different values in the span of 1 to 10 and we found $f = g = 2$ as a good choice. We also checked responses of α, β against *f*, *g*, by fixing *f*, *g* while varying α, β and vice-versa on the scale of 1 to 10. The recorded responses are: (i) eACO algorithm performs best for values $f = g \geq 2$ and $\alpha = \beta = 5$(ii) optimal for $f = g \geq 2$ and $\alpha = \beta = 4, 6$ and (iii) near optimal for $\alpha = \beta \geq 4$ and $f = g = 4,5,6$, as shown in **Fig. 1** for VRPNC_1 instance.

A non-iterative tuning method is used, in which a fixed set of variables is created during initialization only. Then each of these variables is tested in the test phase in order to find the best value in the given set. Hence, this type of tuning follows INITIALIZES and TEST method [47]. Initialization can be done by random sampling, or by generating a systematic grid.

All test instances, were simulated using $\sigma = 6$ elitist ants, which further contributed to update pheromones. An overview of parameters used by eACO algorithm for evaluating CVRP instances is given in Table 2.

Table 2. Parameters used in implementation.

Population size	*M = N-1, customers in each set*
Nearest neighborhood of each city	$NN = N/4$
Initial pheromones	$T_0 = 1.0$ and $T_1 = 1.2$
Alpha and Beta	$\alpha = \beta = 5$
Max_Count	$K = 20$
Elitist ant	$\sigma = 6$
Trail persistence	$\rho = 0.9$
Max_R	$R = 20$
Other scaling parameters	$f = g = 2$, $a = 10$, $b = 10$,
Number of iterations	*Max_Iteration = 500*

In this, Swap heuristic improves the clusters of the solution by changing two cities from different tours. Also, 2-opt is applied to each of the vehicle tour built by the ants, that crosses over itself and reorder it to avoid crossing. For better testing and comparison of all the instances, the maximum iterations are taken as 500.

From computation, it is noticed that the proposed ACO with the above parameters setting is able to achieve optimal solutions in first 200 iterations and also gives a good compromise between solution quality and computation time.

4.3 Computational Results

The algorithm first tested on *set A* and *set B* datasets. Set A consists of 27 instances with maximum of 80 customers, having general type demands and Euclidian distances. On the other hand set B contains 23 instances with number of customers ranging from 31 to 78. **Figure 2** shows the plots for 32 nodes instance of set A and in **Fig. 3** graph for 62 nodes problem of set B is shown.

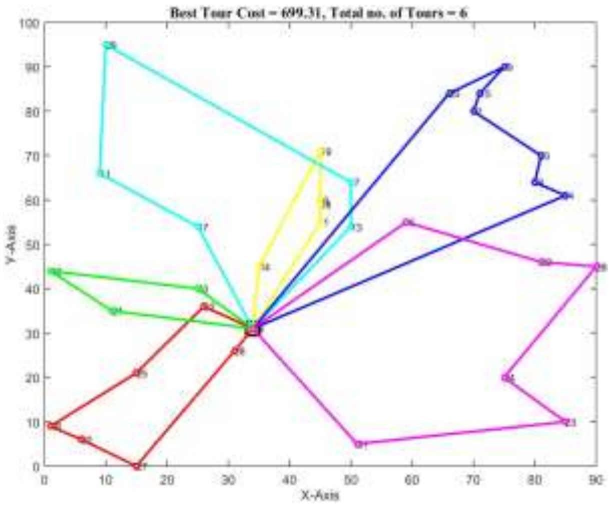

Figure 2. Instance: A_n33_k6, Customers:32, Capacity: 100, Vehicles: 6, Solution: 699.31

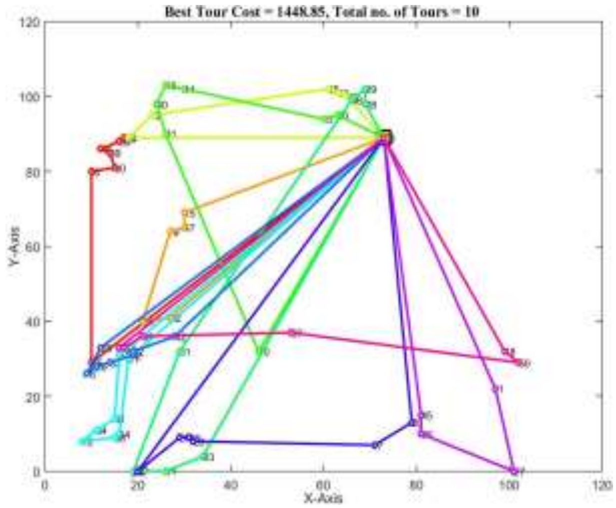

Figure 3. Instance:B_n63_k10, Customers:62, Capacity: 100, Vehicles: 10, Solution: 1448.85

```
Capacity of each vehicle is : 100
Best tour cost is : 699.31
Routes Traversed by Ants are :
Route 1 :  Demand : 97   No. of Customers served : 6
    33    28    27    30    16    25    32    33
Route 2 :  Demand : 86   No. of Customers served : 5
    33    14    19     6    18     1    33
Route 3 :  Demand : 92   No. of Customers served : 3
    33    10    12    21    33
Route 4 :  Demand : 100   No. of Customers served : 5
    33    17    11    29     7    13    33
Route 5 :  Demand : 92   No. of Customers served : 7
    33    20     9    15     2     3     8     4    33
Route 6 :  Demand : 74   No. of Customers served : 6
    33    31    23    24    26    22     5    33
Total Number of Customers Served by all vehicles : 32
```

Figure 4.Routes traversed by the six vehicles for A_n33_k6 instance.

The eACO algorithm has three functions for constraints checking and the newly constructed route is checked for constraint satisfaction by all the three functions. As the algorithm runs, *solution_construction* function builds new route and calculates the sum of demands for all customers belonging to that route. Then the second function, viz., *decision_rule*, checks for capacity and route length constraints by comparing it with vehicle capacity and maximum route length. If for any route, sum of demands exceeds the vehicle capacity, that route will be reconstructed otherwise will be checked for further improvement by *improve_solution* function. Hence, for all routes this 3-step verification is done in order to build the final routes for the customers.

In the following figures for every route, total number of customers' served and their sum of demands is displayed and these demands are either less than or equal to vehicles' capacity in order to follow capacity constraint.

Figure 4 and **5** describe the routes traversed by each of the vehicle to find optimal solutions for the above mentioned datasets. The problem further tested was *set F* (Fisher) dataset, which consists of depot and nodes coordinates, and the customers are separated by EUC_2D type distances. Plots for set F instances for nodes 44 and 134 are shown in **Fig. 6** and **7** respectively.

```
Capacity of each vehicle is : 100
Best tour cost is : 1448.85
Routes Traversed by Ants are :
Route 1 :  Demand : 96   No. of Customers served : 7
    63   49   56   46   59   30   8   35   63
Route 2 :  Demand : 100   No. of Customers served : 5
    63   15   47   9   34   22   63
Route 3 :  Demand : 91   No. of Customers served : 5
    63   32   13   55   3   4   63
Route 4 :  Demand : 100   No. of Customers served : 7
    63   10   31   40   16   44   5   60   63
Route 5 :  Demand : 91   No. of Customers served : 8
    63   28   29   48   61   43   24   11   23   63
Route 6 :  Demand : 96   No. of Customers served : 7
    63   21   14   58   12   54   41   51   63
Route 7 :  Demand : 95   No. of Customers served : 7
    63   25   6   26   33   2   62   42   63
Route 8 :  Demand : 88   No. of Customers served : 6
    63   17   57   38   53   7   20   63
Route 9 :  Demand : 93   No. of Customers served : 4
    63   45   36   27   1   63
Route 10 :  Demand : 72   No. of Customers served : 6
    63   18   50   37   39   52   19   63
Total Number of Customers Served by all vehicles : 62
```

Figure 5. Routes followed by the ten vehicles for B_n63_k10 instance.

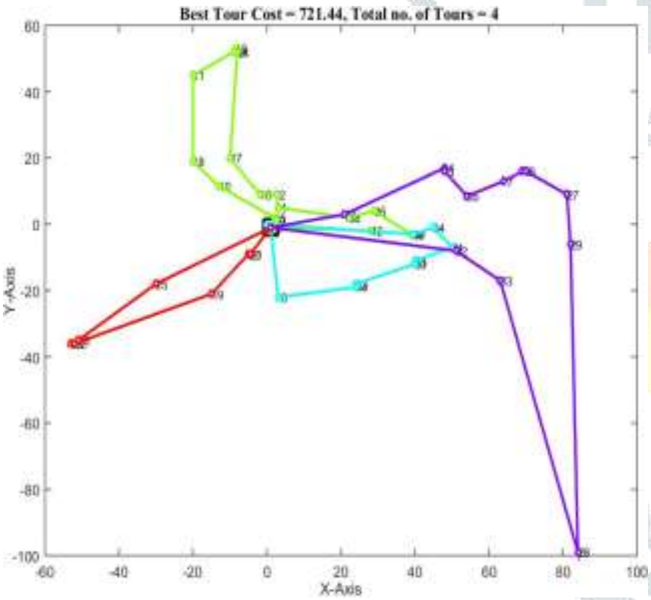

Figure 6. Instance:F_n45_k4, Customers:44
Capacity: 2010, Vehicles: 4,Solution: 721.44

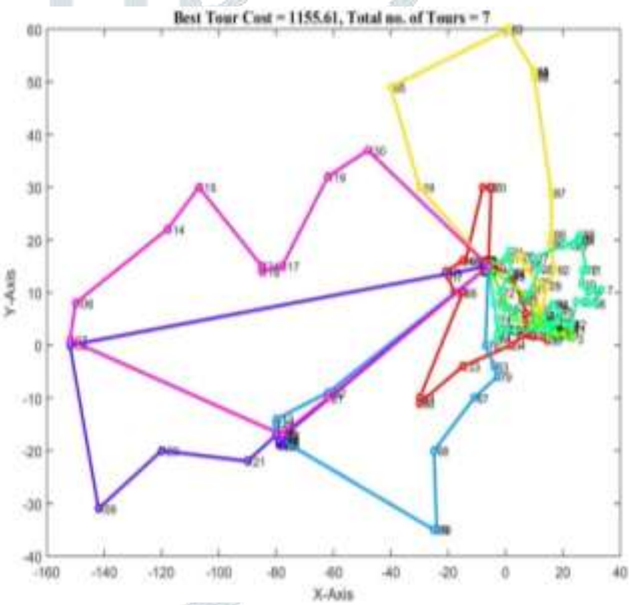

Figure 7. Instance:F_n135_k7, Customers:134,
Capacity: 2210, Vehicles:7, Solution:1155.61

Figure 8 and **9** below shows the route taken by each vehicle along with the demands satisfied for every route for Fisher instances: F_n45_k4 and F_n135_k7 respectively.

```
Capacity of each vehicle is : 2010
Best tour cost is : 721.44
Routes Traversed by Ants are :
Route 1 :  Demand : 1575   No. of Customers served : 7
 45   25   26   23   22   19   20   21   45
Route 2 :  Demand : 1973   No. of Customers served : 16
 45   42   39   36   38   1   2   16   17   14   13   12   11   18   10   15   9   45
Route 3 :  Demand : 1691   No. of Customers served : 9
 45   24   40   34   31   41   30   43   44   8   45
Route 4 :  Demand : 1981   No. of Customers served : 12
 45   37   4   3   35   7   5   6   27   29   28   33   32   45
Total Number of Customers Served by all vehicles : 44
```

Figure 8. Optimal routes for all the four vehicles for F_n45_k4 instance.

```
Capacity of each vehicle is : 2210
Best tour cost is : 1155.61
Routes Traversed by Ants are :
Route 1 :  Demand : 2088   No. of Customers served : 21
  135  20  82  46  118  18  17  71  66  33  80  133  34  49  35  101  50  62  61  60  23  24  135
Route 2 :  Demand : 2147   No. of Customers served : 16
  135  25  26  30  58  57  29  92  90  89  87  86  84  85  83  65  19  135
Route 3 :  Demand : 1904   No. of Customers served : 17
  135  59  55  56  44  41  3  40  95  37  36  104  103  53  54  51  52  1  135
Route 4 :  Demand : 2151   No. of Customers served : 46
  135  91  21  22  72  47  75  31  28  27  16  13  15  88  14  11  12  10  9  8  7  6  5  4
  2  42  43  45  94  93  105  97  96  38  39  98  99  100  102  48  32  134  76  64  77  74  73  135
Route 5 :  Demand : 2189   No. of Customers served : 12
  135  78  63  79  67  68  70  69  111  128  129  113  132  135
Route 6 :  Demand : 2073   No. of Customers served : 11
  135  125  110  122  123  124  126  127  121  120  109  108  135
Route 7 :  Demand : 2068   No. of Customers served : 11
  135  130  119  117  116  131  115  114  106  107  112  81  135
Total Number of Customers Served by all vehicles : 134
```

Figure 9. Optimal routes for all the seven vehicles for F_n135_k7 instance.

The algorithm was finally tested on *set E* (Christofides and Eilon) and CMT (Christofides Mingozzi and Toth) instances. Set *E* consists of total eleven instances and CMT dataset has 14 different types of instances.

Figure 10 and **11** show the MATLAB plots for VRPNC_9, 150 nodes instance of CMT and E_n33_k4, 32 nodes instance of set E respectively.

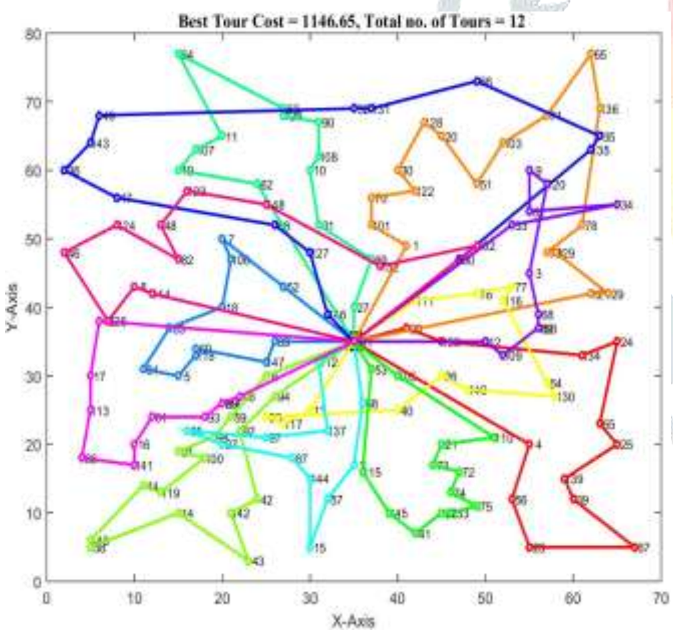

Figure 10. Instance:VRPNC_9,Customers:150 Capacity: 200, Vehicles: 12, Solution: 1146.65

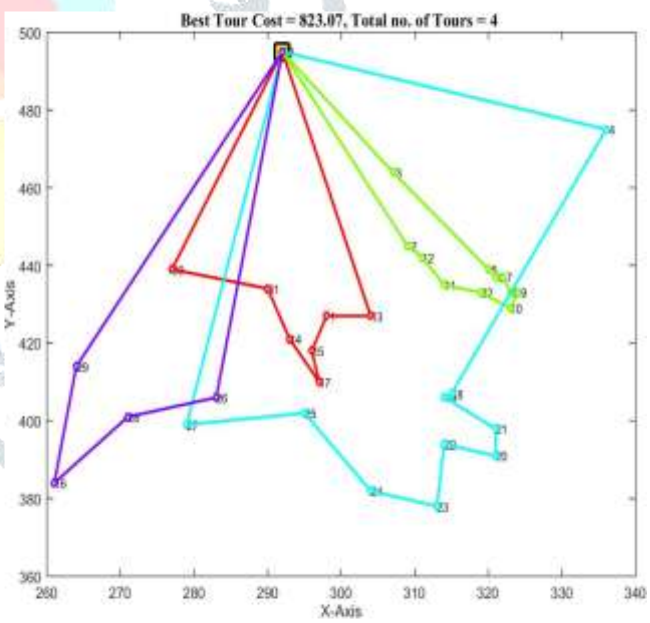

Figure 11. Instance:E_n33_k4, Customers:32, Capacity: 8000, Vehicles: 4, Solution: 823.07

Figure 12 and **13** below show the routes traversed by every vehicle along with the sum of demands fulfilled for each tour, for VRPNC_9 and E_n33_k4 instances respectively.

```
Capacity of each vehicle is : 200
Best tour cost is : 1146.65
Routes Traversed by Ants are :
Route 1 :  Demand : 192  No. of Customers served : 12
  151  28  138  134  24  55  25  139  39  67  23  56  4  151
Route 2 :  Demand : 190  No. of Customers served : 17
  151  1  101  70  122  30  128  20  51  103  71  65  136  78  79  129  29  121  151
Route 3 :  Demand : 188  No. of Customers served : 12
  151  13  117  95  40  26  149  130  54  116  77  76  111  151
Route 4 :  Demand : 183  No. of Customers served : 15
  151  94  92  42  142  43  14  38  140  44  119  100  91  98  59  6  151
Route 5 :  Demand : 177  No. of Customers served : 13
  151  53  115  145  41  22  133  75  74  72  73  21  110  105  151
Route 6 :  Demand : 184  No. of Customers served : 13
  151  62  19  107  11  64  63  126  90  108  10  31  69  27  151
Route 7 :  Demand : 181  No. of Customers served : 11
  151  112  137  97  85  37  87  144  15  57  2  58  151
Route 8 :  Demand : 169  No. of Customers served : 11
  151  52  7  106  18  83  84  5  118  60  147  89  151
Route 9 :  Demand : 199  No. of Customers served : 13
  151  50  135  35  66  131  32  49  143  36  47  88  127  146  151
Route 10 :  Demand : 190  No. of Customers served : 11
  151  12  109  150  80  68  3  120  9  81  34  33  151
Route 11 :  Demand : 186  No. of Customers served : 11
  151  96  99  104  93  61  16  141  86  113  17  45  151
Route 12 :  Demand : 196  No. of Customers served : 11
  151  114  8  125  46  124  82  48  123  148  132  102  151
Total Number of Customers Served by all vehicles : 150
```

Figure 12. Routes traversed by the twelve vehicles for VRPNC_9 instance.

```
Capacity of each vehicle is : 8000
Best tour cost is : 823.07
Routes Traversed by Ants are :
Route 1 :  Demand : 7750  No. of Customers served : 7
  33  30  31  14  17  15  1  13  33
Route 2 :  Demand : 7920  No. of Customers served : 11
  33  2  12  11  32  10  9  8  7  6  5  3  33
Route 3 :  Demand : 7500  No. of Customers served : 10
  33  4  18  19  21  20  22  23  24  25  27  33
Route 4 :  Demand : 6200  No. of Customers served : 4
  33  29  16  28  26  33
Total Number of Customers Served by all vehicles : 32
```

Figure 13. Routes taken by the four vehicles for E_n33_k4 instance.

Figure 14 gives the cost versus iterations plot for Vrpnc_1 instance, which clearly shows that optimal value is achieved after some 130 iterations and after that it remains constant.

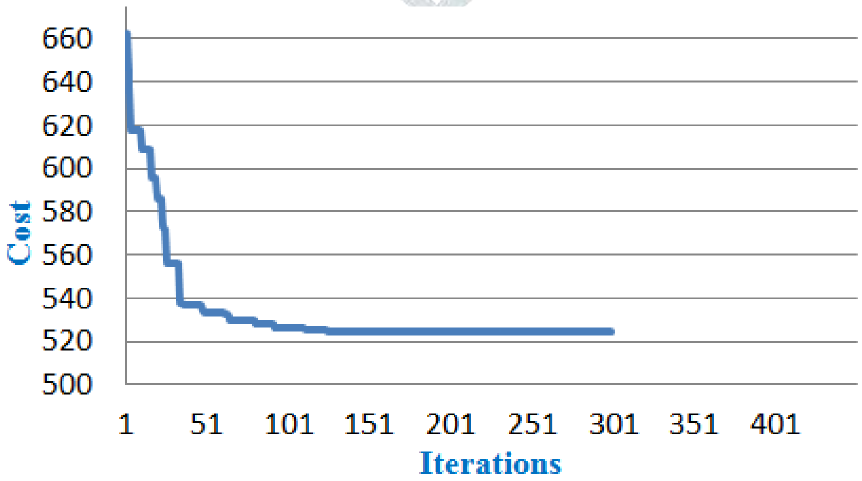

Figure 14. Number of Iterations v/s Cost (524.61) plot for VRPNC_1.

4.4 Results Comparison

This section gives the detailed description about the results obtained for all the benchmarks taken into consideration and also presents the comparison of the results with the best known and other heuristics results given in the literature. The computational results are presented for both best solution obtained and on average solution for 5 runs, for each instance. Table 3 presents the results for Fisher - set F benchmarks.

Table 3. Comparison of present eACO results with best known and other heuristics results for set F

Instance	Best known	CPSO-SA[A]	PACO[B]	PSO-SR-2[C]	LB Tabu[D]	eACO[E]	eACO Avg.[F]	eACO [*min*]
F_n45_k4	724	724	724	724	724	721.44	722.62	1.40
F_n72_k4	237	237	237	237	232.5	238.16	244.27	7.26
F_n135_k7	1162	1200	1170	1162	1157.5	1155.61	1158.46	13.37

[A] Results from Chen (2011) hybrid PSO [38].
[C] Results from Kachitvichyanukul (2009) PSO [18].
[E]Best solution obtained from ACO (present method).
[B] Results from Ting (2012) hybrid ACO& PSO [39].
[D]Results from Augerat et.al.(1998), Tabu Search [40].
[F]Average solution obtained from ACO in 5 runs.

Table 4.Comparison of present eACO results with Best known results and other heuristics for set A

Instance	Best known	PSO[A]	SC-ESA[B]	DELS[C]	SAMCSA[D]	eACO[E]	eACO Avg.[F]	eACO [*min*]
A_n32_k5	784	784	784	784	771.47	**751.49**	**760.07**	*0.52*
A_n33_k5	661	661	661	661	647.48	662.26	668.81	*0.46*
A_n33_k6	742	742	742	742	733.43	**699.31**	**704.73**	*0.42*
A_n34_k5	778	778	778	778	775.95	780.93	784.17	*1.14*
A_n36_k5	799	799	799	799	781.35	802.40	804.38	*1.51*
A_n37_k5	669	669	669	669	673.57	670.02	673.05	*3.18*
A_n37_k6	949	949	949	949	905.98	**930.64**	**936.58**	*3.39*
A_n38_k5	730	730	730	730	716.15	**683.08**	**694.23**	*2.24*
A_n39_k5	822	822	822	822	824.44	824.98	830.02	*3.19*
A_n39_k6	831	831	831	831	827.23	**829.69**	835.25	*4.02*
A_n44_k6	937	937	937	937	928.61	938.18	940.38	*5.27*
A_n45_k6	944	944	944	944	917.14	947.23	954.72	*5.40*
A_n45_k7	1146	1146	1146	1146	1148.84	**1042.91**	**1065.18**	*4.46*
A_n46_k7	914	914	914	914	892.51	**891.07**	**902.27**	*5.33*
A_n48_k7	1073	1073	1084	1073	1064.61	1088.06	1097.64	*5.57*
A_n53_k7	1010	1014	1011	1010	1014.15	**993.53**	1010.12	*6.05*
A_n54_k7	1167	1170	1168	1167	1162.11	**1097.28**	**1122.20**	*6.29*
A_n55_k9	1073	1073	1073	1073	1076.55	**1071.98**	1078.04	*6.52*
A_n60_k9	1354	1356	1355	1354	-	1356.84	1358.46	*6.58*
A_n61_k9	1034	1038	1034	1035	1033.58	1047.46	1056.79	*7.15*
A_n62_k8	1288	1288	1298	1288	-	**1236.94**	**1257.13**	*7.36*
A_n63_k9	1616	1626	1624	1624	-	1634.54	1646.35	*7.02*
A_n63_k10	1314	1320	1315	1316	-	1328.78	1340.37	*7.48*
A_n64_k9	1401	1409	1409	1416	-	1414.75	1428.05	*8.04*
A_n65_k9	1174	1177	1178	1181	1182.21	1191.34	1193.04	*8.22*
A_n69_k9	1159	1162	1159	1165	-	**1158.96**	1172.07	*8.53*
A_n80_k10	1763	1778	1776	1769	-	**1758.69**	1774.69	*9.47*

[A]Chandramouli et. al.(2012) PSO method [34].
[C]Teohet. al. (2015)differential evolution [36].
[E]Solutions obtained by present ACO.
[B] Stanojevic et.al. (2013) savings method [35].
[D]Ernesto et.al. (2014) simulated annealing [37].
[F]Average solutions obtained by our ACO in 5 runs.

Table 3 to 7 report the comparison of results between eACO algorithm and other algorithms for set F, set A, set B, set E and CMT instances respectively. In every table the primary column signifies the CVRP instance name. The second column shows the best-so-far solution for each instance and further columns exhibit the computational results of other existing algorithms. The second-to-last and third-to-last columns represent the results obtained from the proposed ACO method and the average solution computed for 5 runs, respectively. The naming convention is defined for each instance, for example: A_n32_k5 implies instance belongs to set A, having 31 customers and 5 vehicles.

Table 5. Comparison of present eACO results with Best known and other heuristics results for set B

Instance	Best known	PSO[A]	SC-ESA[B]	DELS[C]	SAMC-SA[D]	eACO[E]	eACO Avg.[F]	eACO [*min*]
B_n31_k5	672	672	672	672	616.77	**604.61**	614.67	*0.51*
B_n34_k5	788	788	788	788	772.28	**673.03**	692.47	*0.53*

B_n35_k5	955	955	963	955	887.65	**856.83**	889.26	*1.06*
B_n38_k6	805	805	815	805	691.79	**781.33**	794.79	*1.27*
B_n39_k5	549	549	549	549	539.56	553.15	553.21	*1.42*
B_n41_k6	829	829	866	829	798.42	**819.83**	827.08	*1.58*
B_n43_k6	742	742	746	742	721.61	**728.82**	737.31	*2.23*
B_n44_k7	909	909	921	909	848.60	**870.71**	897.13	*2.46*
B_n45_k5	751	751	751	751	707.08	755.87	764.81	*2.57*
B_n45_k6	678	678	686	678	666.38	686.56	688.05	*3.16*
B_n50_k7	741	741	741	741	685.60	**680.17**	689.40	*2.59*
B_n50_k8	1312	1315	1329	1313	-	**1299.81**	1307.15	*3.37*
B_n51_k7	1032	1038	1032	1033	1007.75	**1031.23**	1033.28	*3.28*
B_n52_k7	747	747	752	747	694.89	**679.17**	684.34	*2.57*
B_n56_k7	707	707	707	707	635.08	**691.66**	699.33	*3.18*
B_n57_k7	1144	1162	1155	1166	1141.53	1173.81	1182.70	*3.56*
B_n57_k9	1598	1598	1600	1599	1524.67	1613.47	1621.87	*3.28*
B_n63_k10	1496	1496	1538	1504	1511.30	**1448.85**	1462.65	*4.35*
B_n64_k9	861	864	861	861	839.17	867.63	874.89	*4.49*
B_n66_k9	1316	-	1341	1322	-	**1308.86**	1315.06	*5.26*
B_n67_k10	1032	1034	1050	1032	1032.37	1049.09	1053.48	*6.08*
B_n68_k9	1272	1273	1292	1281	-	1281.36	1290.36	*5.54*
B_n78_k10	1221	1249	1246	1230	-	**1216.71**	1225.56	*7.08*

[A]Chandramouli et. al. (2012) PSO method [34].
[B] Stanojevic et.al. (2013) savings method [35].
[C] Teohet. al. (2015) differential evolution [36].
[D] Ernesto et.al. (2014) simulated annealing [37].
[E]Best solutions obtained frompresent ACO.
[F]Average solutions obtained from ACO in 5 runs.

Table 6.Comparison of present eACO results with Best known and other heuristics results for set E

Instance	Best known	AMC-PA[A]	DELS[B]	BCP[C]	LB Tab-u[D]	eACO[E]	eACO Avg.[F]	eACO [*min*]
E_n22_k4	375	375	375	375	375	**374.83**	375.06	*0.36*
E_n23_k3	569	569	569	569	569	**524.54**	**531.78**	0.41
E_n30_k3	534	534	534	534	508.5	**505.01**	**520.04**	0.58
E_n33_k4	835	869	835	835	833.5	**823.06**	**833.16**	1.16
E_n51_k5	521	587	521	521	514.52	**511.61**	522.82	2.53
E_n76_k7	682	762	689	682	661.25	**667.24**	683.56	7.49
E_n76_k8	735	819	738	735	711.05	**726.93**	735.67	7.37
E_n76_k10	830	921	843	830	789.31	838.90	848.21	8.05
E_n76_k14	1021	1135	1032	1022	-	**1007.34**	**1018.3**	8.32
E_n101_k8	815	916	822	817	796.15	824.10	836.12	11.51
E_n101_k14	1067	1201	1086	1071	-	**1064.47**	1074.94	12.23

[A]Osaba et. al. (2014)Adaptive multi-crossover [41].
[B] Teohet. al. (2015) differential evolution [36].
[C] Fukasawa (2004) Branch-and-Cut-and-Price [42].
[D] Results from Augerat et.al.(1998), Tabu Search [40].
[E]Solutions obtained by present ACO.
[F]Average solutions obtained by our ACO in 5 runs.

Results and their comparison for 14 instances of Christofides Mingozzi and Toth (CMT) dataset is shown is Table 7. The naming format is given as Vrpnc_1, which means CMT instance 1 and this way for other 13 instances. The number of customers in these 14 instances is ranging from 50 to 199. CMT instances Vrpnc_1 to Vrpnc_5, Vrpnc_11 and Vrpnc_12 are with vehicle capacity constraints, whereas Vrpnc_6 to Vrpnc_10, Vrpnc_13, Vrpnc_14 add an extra restriction of maximum route length and with drop time.

Table 7. Comparison of present eACO results with Best known and other heuristics results for Christofides (CMT) instances

Instance	Best known	Veh. no.[A]	SEP-AS[B]	AGES[C]	OCGA[D]	Veh. no.[E]	eACO[F]	eACO Avg.[G]	eACO [*min*]
Vrpnc_1	524.61	5	524.61	524.61	524.61	5	**524.61**	**524.61**	*2.40*
Vrpnc_2	835.26	10	835.26	835.26	835.26	10	835.84	837.12	*7.35*
Vrpnc_3	826.14	8	826.14	826.14	826.14	8	**826.11**	**826.14**	*12.06*
Vrpnc_4	1028.42	12	1028.42	1028.42	1028.42	12	1028.76	1030.54	*16.53*
Vrpnc_5	1291.29	17	1311.48	1291.29	1299.64	17	1301.3	1308.89	*24.17*
Vrpnc_6	555.43	6	555.43	555.43	555.43	**5**	**533.00**	**538.45**	*2.54*
Vrpnc_7	909.68	11	909.68	909.68	909.68	11	**854.17**	**866.70**	*7.48*
Vrpnc_8	865.94	9	865.94	865.94	865.94	**8**	868.81	874.96	*12.29*
Vrpnc_9	1162.55	14	1162.55	1162.55	1163.38	**12**	**1146.64**	**1152.02**	*17.24*
Vrpnc_10	1395.85	18	1407.21	1401.12	1406.23	**17**	1418.91	1422.54	*25.15*
Vrpnc_11	1042.11	7	1042.11	1042.11	1042.11	7	1045.49	1051.38	*14.22*

Vrpnc_12	819.56	10	819.56	819.56	819.56	10	**819.02**	821.31	*12.57*
Vrpnc_13	1541.14	11	1544.01	1541.14	1542.25	**10**	**1537.52**	1541.85	*14.43*
Vrpnc_14	866.37	11	866.37	866.37	866.37	**10**	**864.46**	868.94	*12.51*

[A]Vehicle used for CMT instances by other heuristics .
[C]Mester(2007) active guided evolution methods [45]
[E]Vehicle used for CMT instances by our ACO.
[G]Average solutions obtained by our ACO in 5 runs.
[B]Tarantilis(2005) Adaptive memory programming [44].
[D]Habibeh(2012) Optimized crossover GA [46].
[F]Solutions obtained by present ACO approach.

The last column of every table represents the computational time (in minutes) taken by the eACO algorithm, to find optimal results for that particular instance. However, as there are contrasts in the simulation setups and number of iterations of various heuristics, a comparison of execution times is hardly meaningful.

V. Conclusion

In this paper, we presented an enhanced ant colony optimization (eACO) algorithm to solve the CVRP optimally. The present approach is examined on several benchmark instances, such as: set A, set B, set E, set F and CMT instances. Results comparisons of our eACO algorithm with other existing heuristics are shown through tables and we can clearly see (highlighted values) that for some instances solutions obtained are superior to best known solutions. We have tested total 78 instances from 5 benchmark problems and we consider the behaviour of eACO very satisfactory, as it is able to improve up to 46 instances and the solution for other 32 instances is close to existing best ones. Likewise, eACO algorithm also reduced the vehicle count for 6 instances of CMT dataset.

The computational results prove that the proposed algorithm is an interesting novel approach to optimize CVRP and can obtain much better solutions in comparison to other existing heuristics. Hence, the most significant contribution of the proposed solution algorithm is its efficiency to optimize both small and large problems of CVRP, both in terms of cost and vehicle count. The proposed eACO algorithm is able to achieve results better than best known results. Still for some of the instances, solutions obtained are not optimal but are near optimal to the best known results. However, further parameter tuning and use of other local search techniques may help to achieve optimum results for all instances.

As for future scope, it will be interesting to improve the performance of the algorithm by parallel implementation along with integration and hybridization of eACO with other intelligent techniques. Besides this, eACO can be tested on other VRP variants such as VRPTW, VRPPD and also eACO results will be compared with our other solution algorithms for VRPs specially PSO and IWD etc.

References

[1] Toth, P., Vigo, D. 2003. The granular tabu search and its application to the vehicle routing problem. Journal on Computing, 15: 333–346.
[2] Gambardella, L. M., Taillard, E., Agazzi, G. 1999. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. [ed.] M. Dorigo, F. Glover D. Corne. New Ideas in Optimization. McGraw-Hill, UK, 63-76.
[3] Laporte, G., Osman, I. H. 1995. Routing problems: A bibliography. Annals of Operation Research, 61: 227-262.
[4] Gutierrez, J. G., Desaulniers, G., Laporte, G., Marianov, V. 2010. A Branch-and-Price Algorithm for Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows. European Journal of Operation Research, 206 (12): 341-349. doi:10.1016/j.ejor.2010.02.037.
[5] Ropke, S., Cordeau, J.F., Iori, M. and Vigo, D. 2007. Branch-and-Cut-and-Price for Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints. Proceedings of ROUTE, Jekyll Island.
[6] Laporte, G. 1992. The Vehicle Routing Problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 59: 345-358.
[7] Baldacci, R., Toth, P., Vigo, D. 2010. Exact algorithms for routing problems under vehicle capacity constraints. Annals of Operations Research, 175: 213–245.
[8] Osman, I. H.1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research, 41: 421–451.
[9] Cordeau, J.F., Laporte, G., Mercier, A. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. Journal of Operational Research Society, 52: 928–936.
[10] Gendreau, M., Hertz, A., Laporte, G. 1994. A tabu search heuristic for the vehicle routing problem. Management Science, 40: 1276–1290.
[11] Ergun, O., Orlin, J.B., Steele-Feldman, A. 2006 Creating very large scaleneighborhoods out of smaller ones by compounding moves. Journal of Heuristics, 12: 115–140.
[12] Chen, P., Huang, K., Dong, Y. 2010. Iterated variable neighborhood algorithm for capacitated vehicle routing problem. Expert Systems withApplications, 37: 1620–1627.
[13] Alba, E., Dorronsoro, B. 2005. The Exploration/Exploitation Tradeoff in Dynamic Cellular Genetic Algorithms. IEEE Transactions on Evolutionary Computation, 9(2): 126-142.
[14] Berger, J., Barkoui, M. 2003. A new hybrid genetic algorithm for the capacitated vehicle routing problem. Journal of the Operational Research Society, 54: 1254– 1262.
[15] Prins, C. 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31: 1985–2002.
[16] Prins, C. 2009. A GRASP evolutionary local search hybrid for the vehicle routing problem. In: Pereira, F.B., Tavares, J. (Eds.), Bio-inspired Algorithms for the Vehicle Routing Problem. Springer-Verlag, Berlin, Heidelberg, 35–53.
[17] Shen, Y., Murata, T. 2012. Pick-up Scheduling of Two-Dimensional Loading in Vehicle Routing Problem using GA. Lecture Notes in Engineering & Computer Sc., 6(2): 1532-1537.
[18] Ai, J., Kachitvichyanukul, V. 2009. Particle swarm optimization solution representations for capacitated vehicle routing problem. Com-

puters & Industrial Engineering, 56: 380–387.
[19] Kanthavel, K., Prasad, P. 2011. Optimization of Capacitated Vehicle Routing Problem by Nested Particle Swarm Optimization. American Journal of Applied Sciences,8(2): 107-112.
[20] Yin, L., Liu, X. 2009. A Single–depot Complex Vehicle Routing Problem and its PSO Solution. Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT '09) China, 266-269.
[21] Bullnheimer, B., Hartl, R.F., Strauss, C. 1999. An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research, 89: 319–328.
[22] Dorigo, M., Birattari, M., Stutzle, T. 2006. Ant colony optimization. Computational Intelligence Magazine, IEEE, 1: 28-39.
[23] Reimann, M., Doerner, K., Hartl, R. F. 2004. D-Ants: Savings based ants divide and conquer the vehicle routing problem. Computers & Operations Research, 31: 563–591.
[24] Yu, B., Yang, Z.-Z., Yao, B. 2009. An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research, 196: 171– 176.
[25] Marinakis, Y., Marinaki, M., Dounias, G. 2010. Honey bees mating optimization algorithm for large scale vehicle routing problems. Natural Computing, 9: 5–27.
[26] Gendreau, M., Potvin, J.Y., Braysy, O., Hasle, G., Lokketangen, A. 2008. Metaheuristics for the vehicle routing and its extensions: A categorized bibliography. In: Golden, B., Raghavan, S., Wasil, E. (Eds.). The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, 143–169.
[27] Gupta, A., Saini, S. 2017. On Solutions to Vehicle Routing Problems using Swarm Intelligence Techniques: A Review. In: Bhatia S., Mishra K., Tiwari S., Singh V. (eds) Advances in Computer and Computational Sciences. Advances in Intelligent Systems and Computing, Springer, Singapore, 553: 345-354.
[28] Augerat, P., Belenguer, J., Corberan, A., Naddef, D., Rinaldi, G. 1995. Computational results with a branch and cut code for the capacitated vehicle routing problem. Tech. Rep. 949-M, University Joseph Fourier, Grenoble, France.
[29] Christofides, N., Eilon, S. 1996. An algorithm for the vehicledispatching problem.Operational Research Quarterly, 20: 309–318.
[30] Christofides, N., Mingozzi, A., Toth, P. 1995. The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., (Eds.), Combinatorial Optimization (Wiley), 1: 315-338.
[31] Fisher, M. 1994. Optimal solution of vehicle routing problems using minimum K-trees. Operations research, 42(4): 626–642.
[32] Dorigo, M. 1992. Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy.
[33] http://vrp.atd-lab.inf.puc-rio.br/index.php/en/
[34] Chandramouli, A., Narendran, T., Srinivasan, L.V. 2012. Efficient heuristics for large-scale vehicle routing problems using particle swarm optimization. International Journal ofGreen Computing, 3(2): 34–50.
[35] Stanojevic, M., Stanojevic, B., Vujosevic, M. 2013. Enhanced savings calculation and its applications for solving capacitated vehicle routing problem. Applied Mathematics & Computation, 219(20): 10302–10312.
[36] Teoh, B. E., Ponnambalam,S. G., Kanagaraj G. 2015. Differential evolution algorithm with local search for capacitated vehicle routing problem.International Journal of Bio-Inspired Computation, 7(5): 321-342.
[37] Garcia, E., Crystal, L., Salvador,D. 2014. Solving the capacitated vehicle routing problem with stochasticdemand applying the simulated annealing algorithm. ProgramaciónMatemáticay Software, 6(2): 36-45.
[38] Kao, Y., Chen, M. 2011. A hybrid PSO algorithm for the CVRP problem.In Proceedings of the International Conference on Evolutionary Computation Theory & Applications (ECTA '11), Paris, France, 539–543,.
[39] Kao, Y.,Chen, M. H., Huang, T. 2012. A hybrid algorithm based on ACO and PSO for capacitated vehicle routing problems. Mathematical Problems in Engineering, 2012: 17 pages.
[40] Augerat, P.,Belenguer,J.M.,Benavent, E.,Corberan, A.,Naddef, D. 1998. Separating capacity constraints in the CVRP using tabu search. European Journal of Operational Research, 106(2): 546-557.
[41] Osaba, E., Onieva, E., Carballedo, R., Diaz, F., Perallos, A. 2014. An adaptive multi-crossover population algorithm for solving routing problems. In Nature Inspired CooperativeStrategies for Optimization (NICSO) Springer, 113–124.
[42] Fukasawa, R., Lysgaard, J., Werneck, F. R. 2004. Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. Integer Programming and Combinatorial Optimization: 10th International IPCO Conference, Springer, New York, USA, 10: 2-15.
[43] Alba, E., Doronsoro, B. 2006. Computing nine new best-so-far solutions for Capacitated VRP with cellular Genetic Algorithm. Information Processing Letters, Elsevier, 98: 225-230.
[44] Tarantilis, C.D. 2005. Solving the vehicle routing problem with adaptive memory programming methodology, Computers & Operations Research 32: 2309–2327.
[45] Mester, D., Bräysy, O. 2007. Active guided evolution strategies for large-scale capacitated vehicle routing problems. Computers & Operations Research, 34: 2964–2975.
[46] Habibeh, N., Lai, S. L. 2012. Optimised crossover genetic algorithm for capacitated vehicle routing problem. Applied Mathematical Modelling, Elsevier, 36: 2110–2117.
[47] Smit, SK., Eiben, A. 2010. Parameter tuning of evolutionary algorithms: Generalist vs. specialist. In: European Conference on the Applications of Evolutionary Computation. Switzerland: Springer International Publishing, 542–51.
[48] Gupta, A., Saini, S. 2018. On Solutions to Capacitated Vehicle Routing Problem Using an Enhanced Ant Colony Optimization Technique. In: Perez G., Mishra K., Tiwari S., Trivedi M. (eds) Networking Communication and Data Knowledge Engineering. Lecture Notes on Data Engineering and Communications Technologies, Springer, Singapore, 3: 257-266.