

A Hybrid Particle Swarm Optimization with Genetic Operators for Vehicle Routing Problem

S. Geetha

PSG College of Technology, Coimbatore, India
 Email: geet_shan@yahoo.com

G. Poonthalar and P. T. Vanathi

PSG College of Technology, Coimbatore, India
 Email: thalirkathir@rediffmail.com and ptvani@yahoo.com

Abstract—The Vehicle Routing Problem (VRP) is a NP-hard and Combinatorial optimization problem. Combinatorial optimization problem can be viewed as searching for best element in a set of discrete items, which can be solved using search algorithm or meta heuristic. In this work, VRP is solved using population based search algorithm, Particle Swarm Optimization (PSO) with crossover and mutation operators. In this paper, the PSO for VRP is considered from two aspects: the hybrid PSO algorithm and particle encoding method. The technical details required for this application are discussed. The computational result shows that the results of hybrid PSO for VRP are competitive.

Index Terms—Genetic operators, Hybrid Strategy, Particle swarm optimization, Particle encoding, Vehicle Routing Problem.

I. INTRODUCTION

The VRP was first introduced by Dantzig and Ramset [8], where the objective is to find a route with the lowest cost for the vehicles stationed at a depot and to satisfy the demands of customers situated at geographically dispersed location. The route of each vehicle should start and end at the depot by visiting each customer only once. VRP in a broad sense is a generic name given to a whole class of problems in which a set of routes for a fleet of vehicles based at one or several depots. Because of its overall presence in the fields of transportation, distribution and logistics VRP arises as one of the most challenging combinatorial optimization task, even it is defined more than fifty decades ago. Often the context of VRP is to deliver the goods located at central depot to customers who have placed order for goods. The ultimate goal of VRP is to minimize the travel cost of vehicles for distributing the goods to the customer. Figure 1 shows an example of VRP with 9 customers with two routes.

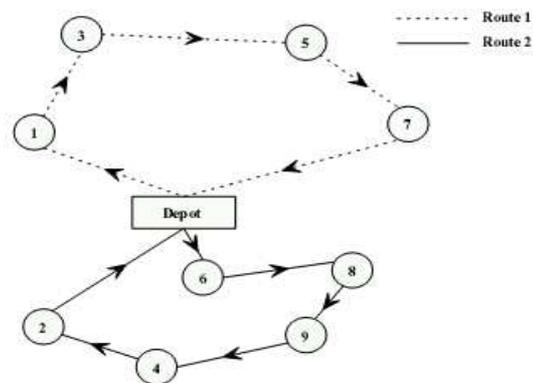


Figure 1. Example of VRP

Several variations of the VRP are existing [12] as follows,

- Capacitated VRP (CVRP) in which the vehicles are limited with capacity while servicing the customer demands.
- VRP with Time Window (VRPTW) where the customers have time windows within which the deliveries must be made.
- VRP with pickup and delivery (VRPPD) is to find optimal route for a fleet of vehicles to visit the pickup location for loading the goods and drop the goods to different delivery location.
- Distance Constraint VRP (DCVRP) has maximum length for each vehicle to travel.

Generally, VRP is divided into two subclasses, asymmetric VRP (aVRP) and symmetric VRP (sVRP). Let the traveling cost from location a to location b is $cost_{ab}$. If the $cost_{ab}$ is equal to $cost_{ba}$ then it is sVRP, otherwise aVRP. This paper involves sVRP using capacitated vehicles.

Many solutions are proposed to solve VRP a NP-hard problem and finding a globally minimum solution is computationally complex. As a consequence,

evolutionary computing methods have been applied for VRP to find a near optimal solution in a reasonable amount of time. For example: genetic algorithm, Ant Colony Optimization and Particle Swarm Optimization.

Particle Swarm Optimization (PSO) is a population based search algorithm proposed by Kennedy and Eberhart [13] which was motivated by the organism behavior such as schooling of fish and flocking of birds. PSO can solve a variety of difficult optimization problems. PSO uses the physical movement of the individuals in the swarm. PSO has a flexible, well-balanced mechanism to enhance and adapt, to the global and local exploration abilities. PSO is simple in coding and consistency in performance.

In order to make PSO applicable to VRP, the relationship between particle position and vehicle routes must be clearly defined. The definition of particle as an encoded solution is usually called a solution representation and the method to convert it to problem specific solution is usually called a decoding method. In this work, permutation encoding is used to represent the solution and it is decoded as routes for each vehicle. Genetic operators are incorporated with PSO for updating the particles.

II. LITERATURE SURVEY

Many solution techniques have been proposed to solve VRP. Some techniques include Tabu Search (TS) [10], Simulated Annealing (SA) [17] etc. In recent years Genetic Algorithm(GA) and PSO have drawn a great deal of attention from researchers due to its robustness and flexibility. Yi-Liang Xu [21] in their paper, proposed three genetic representations for VRP which uses fuel truck dispatch system data set for their investigation with hybrid GA and 2-opt routing optimization technique. GA with neighborhood search method was adopted by Baker et al. [4]. They proved that the results are competitive with TS and SA. In [5], a hybrid genetic algorithm to address the capacitated vehicle routing problem is proposed. A finite automata to produce individual population and a different evolutionary way enlightened by hermaphrodites to VRP was proposed by Y Zhang et al. [22]. A hybrid algorithm based on GA and TS for solving CVRP was studied by Guido et al. [11], which describes two simple heuristics and introduced a new chain mutation genetic operator. Thangiah and Gubbi [19] used GA to find good cluster of customers for VRP with 'cluster first and route second' heuristics.

N.Li [16], Z.J. Shao [18], and X. G. Luo [14] adapt a mixed approach GL-PSO for solving VRP, Vehicle routing with stochastic travel times, vehicle with non-full load respectively. Their results are proved to be quick and effective. N.Z. Zhang [23] applied FDR-PSO (Fitness-Distance-Ratio based PSO) to facility VRP using *nbest* along with *gbest* and *lbest*. In Chen et al [7], discrete particle swarm optimization combines global search, local search and SA with certain probability to avoid

being trapped in local optimum for CVRP. Ai and Kachitvichyanukul [1] used classical PSO without any hybridization. They also used GLNR-PSO for solving VRP with simultaneous pickup and delivery [2] and CVRP [3]. GLNR-PSO is a multiple social structure technique by combining *nbest*, *lbest* and *gbest*. Y. Wu [20] used multi-population PSO for solving VRPTW. More researches are done by incorporating Ant Colony Optimization, Tabu search, etc. in PSO for solving VRP and its variants.

In this work, PSO with genetic operators are used to solve VRP. The paper is organized with brief problem definition in section 3, detailed explanation of proposed work in section 4 with results and discussion in section 5 and 6. Section 7 discusses the computational complexity of proposed work followed by conclusion of work in section 8.

III. PROBLEM DEFINITION

VRP is generally defined as follows: n customers with known demand d_i ($i=1$ to n) are serviced with m vehicles stationed at depot with uniform capacity q . VRP is to service all the customers with the following objectives and constraints:

Objectives

- Minimize the total number of vehicles used
- Minimize the distance traveled by each vehicle

Constraints

- Load of each vehicle should not exceed the given vehicle capacity
- Each customer is serviced exactly once
- Each vehicle route starts and ends at depot

The problem is given with a set of

$$\begin{aligned} \text{Customers} &: c_1, c_2, c_3 \dots c_n \\ \text{Demands} &: d_1, d_2, d_3 \dots d_n \\ \text{Vehicles} &: v_1, v_2, v_3 \dots v_m \\ \text{Capacity} &: q \end{aligned}$$

where $c_i \in C$ are the set of customers distributed in the Euclidean plane (x_i, y_i) whose distances are symmetric, the demand (d_i) and capacity (q) of vehicle are positive integers. The $Cost_{ij}$ is the travelling cost/distance between customer i to customer j .

The objective function is shown as below

$$\min \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n \cos t_{ij} x_{ijk} \quad (1)$$

Subject to

$$\sum_{i=1}^n d_i y_{ki} \leq q, k = 1, 2, \dots, m \quad (2)$$

$$\sum_{k=1}^m y_{ki} = 1, i = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=0}^n x_{ijk} = y_{kj}, j = 0, 1, 2 \dots m, \forall k \quad (4)$$

$$\sum_{j=0}^n x_{ijk} = y_{ki}, i = 0, 1, 2 \dots, n, \forall k \quad (5)$$

Where x_{ijk} is a binary variable indicating whether vehicle k arrives at customer j from customer i , $x_{ijk}=1$ if vehicle k arrives at customer j from customer i , otherwise 0. The y_{ki} is a binary variable indicating whether the customer i is served by vehicle k , $y_{ki}=1$ if customer i is served by vehicle k , otherwise 0. The objective function (1) strives to minimize the total travel cost of each vehicle within a group of customer. The constraint (2) is to restrict that the total demand of the vehicle should not exceed its capacity. Constraint (3) represents each customer demand must be satisfied and fulfilled by one vehicle. Constraint (4) and (5) ensure that only one vehicle arrive and depart from a customer.

IV. PROPOSED WORK

In this work, the PSO is incorporated with crossover and mutation operators of GA to provide better results. Initial particles are generated using greedy approach and updated using varying inertia. Permutation encoding is used to represent the solution. A decoding procedure is used for converting solution to routes. A local search 2-opt is used to optimize the route of each vehicle.

A. Particle Swarm Optimization

Each single solution is a bird, called particle in the search space. All particles have fitness value, evaluated using fitness function and velocities directing the particles towards solution. The dimension of a particle is N and the number of particle is M . The i^{th} particle is represented by $X_i = (x_{i1}, x_{i2} \dots x_{in})$. The best solution is represented as $pbest$ or $P_i = (p_{i1}, p_{i2}, p_{i3} \dots p_{in})$. The velocity rate of the particle is $V_i = (v_{i1}, v_{i2}, v_{i3} \dots v_{in})$. In the process of iteration, particles update their own velocities and positions using the following equations (6) and (7).

$$v_{in}(t+1) = w v_{in}(t) + (c_p \text{ rand}()) [p_{in}(t) - x_{in}(t)] + (c_g \text{ rand}()) [p_{gn}(t) - x_{in}(t)] \quad (6)$$

$$x_{in}(t+1) = x_{in}(t) + v_{in}(t+1) \quad (7)$$

$1 \leq i \leq M, 1 \leq n \leq N$, c_p, c_g are learning factors, $\text{rand}()$ is a random number between (0,1). The w is an inertia factor.

B. Particle Swarm Optimization for VRP with Genetic Operator

The initial solutions are generated using greedy algorithm in step 1. Then the solutions are decoded to the route of each vehicle and its fitness values are evaluated in steps 2 and 3. The solutions are converted to particles

position value [6] in step 4. The particles social and cognitive information are updated in steps 5 and 6 and moved by step 9. The elitism is added to preserve the best particles in step 7. After moving, crossover and mutation are applied for particles with $gbest$ in steps 10 and 11. Particles position values are converted back to solution using Ranked Order Value (ROV) or Smallest Position Value (SPV) [9] in step 12. Step 13 is a controlling step for repeating or stopping the iteration. The algorithm is described as follows.

Algorithm 1. PSO for VRP using genetic operators

1. Initialize I solutions (Y_i) as a population, using greedy method. Set $V_i = 0$ for $i=1, 2, 3 \dots I, t=1$.
2. For $i=1, 2, 3 \dots, I$, decode Y_i to a set of vehicle routes R_i .
3. For $i=1, 2, 3 \dots, I$, evaluate the fitness of R_i . Apply local exchange 2-opt to improve the obtained routes R_i .
4. For $i=1, 2, 3 \dots I$, convert Y_i into X_i , the particles position value
5. Set or update the $pbest, P_i = X_i$, if fitness of $X_i <$ fitness of P_i
6. Set or update the $gbest, P_g = P_i$, if fitness of $P_i <$ fitness of P_g
7. 20% of particles are preserved without updation as elitism
8. Repeat the steps 9, 10 and 11 for remaining 80% of particles
9. Update the velocity and the position of each i^{th} particle:

$$w(t) = w_{max} + (t-T)/(1-T)[w_{min} - w_{max}]$$

$$v_{in}(t+1) = w(t) v_{in}(t) + (c_p \text{ rand}()) [p_{in}(t) - x_{in}(t)] + (c_g \text{ rand}()) [p_{gn}(t) - x_{in}(t)]$$

$$x_{in}(t+1) = x_{in}(t) + v_{in}(t+1)$$

10. Perform single point crossover on X_i with $gbest$
11. Perform swap and inverse mutation on X_i based on the probability p_m
12. Convert the particle position value X_i into solution Y_i using smallest position value for all i
13. If the stopping criterion is met, go to step 2, otherwise $t=t+1$
14. Decode P_g as the best set of vehicles route

C. Solution Representation

The solution is encoded as integer permutation, $Y_i = (y_{i1}, y_{i2}, y_{i3} \dots y_{in})$. The length of the solution is n where each y_{ij} represent the customer. While decoding the solution, the demands are also considered. Each particle is decoded from left, forming a route for each vehicle, satisfying the capacity constraint. The algorithm 2 describes the decoding procedure of solution representation.

Algorithm 2. Decoding method of solution representation

1. Set $k=1$
2. Scan the solution Y_i from left to right for all i , $1 \leq i \leq I$
3. Include the first unvisited customer y_{ij} into the route r_{ik} for $1 \leq i \leq I$, $1 \leq j \leq D$ with the capacity constraint (2)
4. If capacity constraint violated, perform local exchange 2-opt for r_{ik} , increment the k
5. If $k > m$ stop; else go to step 2

The algorithm is explained in the figure 2 with an example of 6(=n) customers and 3(=m) vehicles of uniform capacity 20. The route construction for vehicle 1 starts from y_{i1} i.e. customer 2 onwards. Customer 2 is assigned to vehicle 1, and then customer 3 is assigned. While assigning customer 4, the constraint (2) is not satisfied, so it is not assigned. Then customer 1 which satisfies the constraint (2) is included to route 1 or vehicle 1. Likewise, the procedure is repeated for all vehicles. Figure 2 illustrates the decoding procedure.

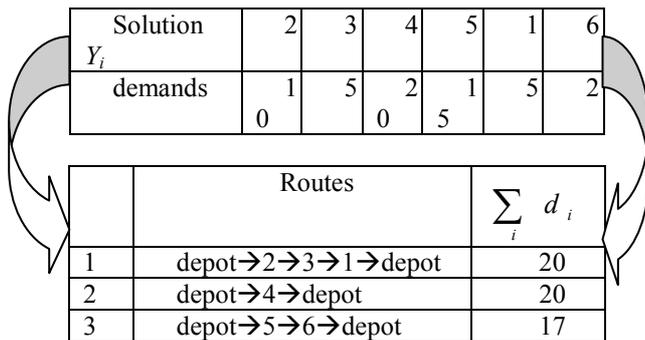


Figure 2. Solution representation and its decoded routes

D. Particle Initialization

In basic PSO, initial particles are generated randomly. To guarantee an initial population with certain quality and diversity, initial solutions are generated using greedy method. Every initial solution generated is a permutation of vehicle routes. It should be converted to particles position value. The conversion is performed using the following equation (8).

$$x_{ij} = x_{min} + (x_{max} - x_{min}) / n [(y_{ij} - 1 + \text{rand}())] \quad (8)$$

y_{ij} is the j^{th} dimension of the i^{th} solution, x_{ij} is the j^{th} dimension of i^{th} particle, $\text{rand}()$ is uniformly distributed in $[0,1]$, n represent the number of customers, x_{min} and x_{max} are the boundary values of a particle position.

For 5 customers, let $Y_i = \{1\ 2\ 3\ 4\ 5\}$ be the permutation encoding of a solution. Each y_{ij} is converted to a particle position value x_{ij} using (8) is shown in the following figure 3.

Dimension j	1	2	3	4	5
Permutation encoding (Y_i)	1	2	3	4	5
Position value (X_i)	0.304	0.582	0.807	1.528	1.777

Figure 3. Conversion of solution to a particle

These continuous particle position values $X_i = [x_{i1}, x_{i2}, x_{i3} \dots x_{id}]$ are converted back to permutation of solution $Y_i = [y_{i1}, y_{i2}, y_{i3} \dots y_{id}]$ using ROV. The ROV uses Smallest Position Value (SPV) of a particle and assigns a smallest rank value 1. Then the next SPV is assigned as 2. In the same way all SPV of particles are handled and assigned with the rank value to a route permutation. The figure 4 illustrates the ROV of a particle to a permutation encoding. The SPV is $x_{i1} = 0.380$, the y_{i1} is assigned the rank value 1, then the next SPV is $x_{i3} = 0.642$, y_{i3} is assigned with value 2. Repeating the same, the ROV Y_i is obtained as $\{1\ 4\ 2\ 3\ 5\}$.

Dimension j	1	2	3	4	5
Position value (X_i)	0.380	1.292	0.642	0.994	1.956
ROV	1	4	2	3	5

Figure 4. Representation of Particle and ROV

E. Genetic Operators

The crossover and mutation [15] are the two operators applied for particles to evolve better in next iteration. Single point cross over is applied to all particles with the g_{best} , (P_g). This is done in order to move the particle towards the global optima. The particles are mutated based on P_m . Swap mutation is applied if the random number is less than the P_m , otherwise inversion mutation is applied. The crossover and mutation are elaborated in Algorithm 3.

Algorithm 3. Genetic Operators, Crossover and Mutation

1. For each particle X_i
 - a. Select a random point p between 1 to n
 - b. Swap the particles value with P_g , i.e. $x_{ip}, x_{ip+1}, \dots, x_{in}$ with $P_{gp}, P_{gp+1}, \dots, P_{gn}$
 - c. If $\text{rand}() < P_m$, then two random point is selected p_1, p_2 to perform swap mutation i.e. to swap x_{ip1} and x_{ip2}
 - d. If $\text{rand}() > P_m$, inversion mutation is done by swapping all positions from x_{ip1} to x_{ip2}

F. Local Exchange 2-opt

The common local improvement procedure 2-opt is used in the decoding method to improve the routes of each vehicle. A 2-opt move consists of eliminating two customers and reconnecting the two resulting paths in a different way to obtain a new route. Among all pairs of

customer whose 2-opt exchange decreases the length; choose the pair that gives the shortest route. This procedure is then iterated until no such pair of customers is found. The resulting route is called 2-optimal. The formal algorithm for 2-opt is explained in algorithm 4.

Algorithm 4. 2-opt local exchange

1. For $i=1,2,3,\dots,n$ and $j=1,2,3\dots m$
2. Set n_j =number of customers in the route r_{ij}
3. For $p=1,2,\dots,(n-2)$ and $q=(p+2)\dots n$
 - 3.1 Modify route by changing the route direction of customer in the sequence number $p,p+1$ and $q,q+1$
 - 3.2 Evaluate the routing cost
 - 3.3 If improves modify the route, else return the route without modification

V. COMPUTATIONAL RESULTS

The proposed algorithm is tested with Vehicle Routing Problem (VRP) Bench mark dataset of Augerat, et al set A, Augerat, et al set B, Augerat, et al set P, Christofides and Eilon.

The various form of PSO like simple PSO, PSO without local exchange, hybrid PSO with crossover and mutation algorithms have been implemented in MATLAB 7.0.1 with Pentium 4.0, 2.3 GHz and the algorithm is tested with 10 problem instances of the benchmark data sets. The parameters of PSO: Number of Particles $I = 100$, Number of Iteration $T=100$, Initial inertia weight $w_{min} = 0.9$, Last inertia weight $w_{max}=0.1$, personal best position acceleration constant $c_p=0.5$, Global best position acceleration constant $c_g=0.5$. The mutation probability rate $p_m=0.4$ for performing swap or inversion mutation. The elitism of 20% is used for updating the particle. The boundary values $x_{min}=0.1$ and $x_{max}=0.7$ are used for converting the solution to a particle. In the benchmark data set each customer has a node number and a demand. Every customer has x and y coordinates and are randomly distributed in the Euclidean plane. The Table 1 lists the characteristics of the problem sets. The column specifies the problem instance, number of customers, vehicle capacity, and number of vehicles respectively. The problem instances of large number of customers, high vehicle capacity, customers with high demands are tested and the results are tabulated in Table 2. The data set is available at <http://www.branchandcut.org>.

TABLE I. CHARACTERISTICS OF PROBLEM SET

Problem instance	No. customers	Vehicle capacity	No. Vehicles
A-n33-k5	32	100	5
A-n46-k7	45	100	7
A-n60-k9	59	100	9
B-n35-k5	32	100	5
B-n45-k5	44	100	5

B-n68-k9	67	100	9
E-n30-k3	29	4500	3
E-n51-k5	50	160	5
F-n72-k4	71	30000	4
P-n101-k4	100	350	4

The computational results for problem instances are projected in table 2, with Best Known Solution and solution obtained by Chen [7] and Ai et al (SR-2) [3].

TABLE II. RESULTS OBTAINED FOR BENCHMARK DATA SET

Problem instance	BKS	Chen [7]	Ai (SR-2) [3]	Obtained solution
A-n33-k5	661	661	661	661
A-n46-k7	914	914	914	921
A-n60-k9	1354	1354	1355	1368
B-n35-k5	955	955	955	955
B-n45-k5	751	751	751	754
B-n68-k9	1272	1272	1274	1281
E-n30-k3	534	534	534	534
E-n51-k5	521	528	521	522
F-n72-k4	237	244	237	253
P-n101-k4	593	602	594	610

The initial best solution and final best solution obtained for the problem instances are projected with the improvement rate in Table 3. The improvement rate is calculated as

$$\text{Improvement rate} = \frac{\text{Initial best solution} - \text{Final best solution}}{\text{Initial best solution}} \times 100\%$$

TABLE III. IMPROVEMENT RATE FOR PROBLEM INSTANCES

Problem instance	Initial Best Solution	Final Best Solution	Improvement rate (%)
A-n33-k5	720	661	8.19
A-n46-k7	1102	921	16.42
A-n60-k9	1617	1368	15.39
B-n35-k5	1062	955	10.07
B-n45-k5	820	754	8.04
B-n68-k9	1465	1281	12.55
E-n30-k3	613	534	12.88
E-n51-k5	657	522	20.54
F-n72-k4	350	253	27.71
P-n101-k4	1016	610	38.26

The work is also executed as a simple PSO without including GA operators. But the results are not converging to (near) optimal.

VI. DISCUSSIONS

The results of permutation encoding are shown in the table 2 which is challenging with BKS, Chen and Ai et al results. The proposed work fails to meet the optimal route

length, when the problem instances are big; i.e. having customers of 60 and above. For other problem instances it is almost close to BKS. The n -dimensional particles are used in this work improves the storage efficiency and also runs faster when compared to $n+2m$ or $3m$ -dimensional particle[3].

The table 3 shows the improvement rate of HPSO. The initial cost of all problem instances are less because of the inclusion greedy algorithm and 2-opt exchange. If greedy algorithm is not included then the initial solution is of so high that takes probably more iteration to converge or may not converge. Figure 5 shows the initial cost is 1220 for a-n33-k5 which takes more than 100 iterations to converge. The improvement rate is less because of the use of heuristics such as greedy and 2-opt exchange. If these heuristics are not used then the initial solution is high, which may increase the improvement rate but will not assure quick converges. Crossover and mutation makes the exploration of various routes in the large solution space.

The effectiveness of hybrid PSO algorithm is shown by comparing the cost with number of iterations. The results are projected for the problem instance a-n33-k5 whose optimal route cost is 661. Fig 5 projects the cost for PSO algorithm without greedy for its initialization. The initial cost is 1220 and even after 100 iterations, it was not converging to 661.

Figure 6 shows the results obtained for PSO with greedy, but without local exchanges. The initial cost is 726 since greedy is used for initialization, because of which it converges within 200 iterations, to an optimal value.

The cost obtained by Hybrid PSO using greedy as well as 2-opt method are shown in fig 7. The initial cost is 726 and because of local exchange an optimal solution is obtained within 100 iterations.

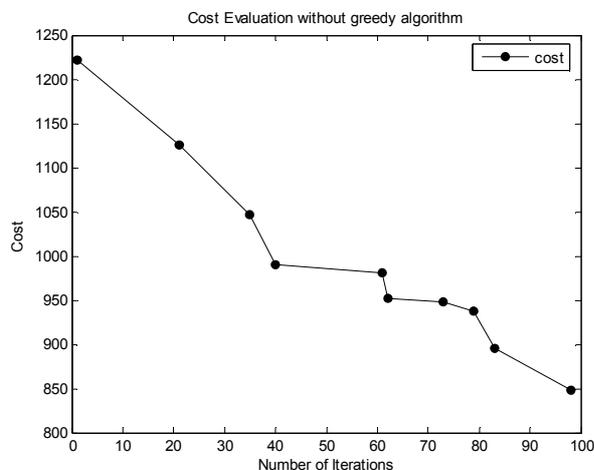


Figure 5. Cost Evaluation without greedy algorithm

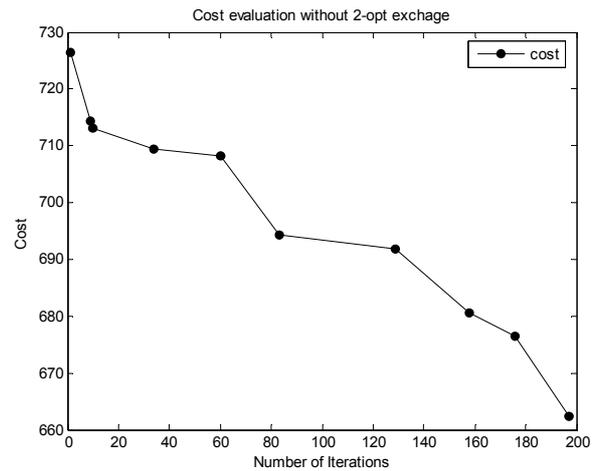


Figure 6. Cost Evaluation without local exchange

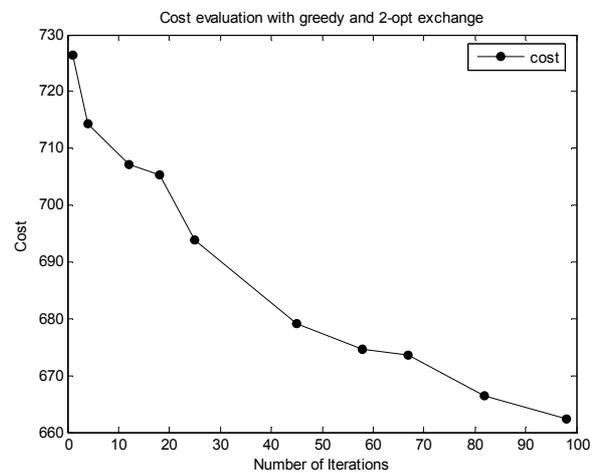


Figure 7. Cost Evaluation for Hybrid PSO

VII. COMPUTATIONAL COMPLEXITY

The NP-hard VRP is decomposed into subtask as detailed in section 4, which are relatively solvable tasks. Let n be the number of customers and m be the number of vehicles in a depot to service the customers. Then the space complexity is analyzed based on the size of the particles used by PSO. The space complexity is projected in the table 4.

TABLE IV SPACE COMPLEXITY

Algorithm	Particle Size
Chen	nm
Ai et al.	$n+2m(SR1)$ $3m(SR2)$
Proposed Algorithm	n

The size of particle in the proposed work is n , which automatically reduces the space needed for storing as well as the time taken to decode and for other operations. The time complexity is of the proposed work is given in the table 5.

TABLE V TIME COMPLEXITY

Algorithm	Complexity of an algorithm
Decoding Algorithm	$O(nm)$
Particle Conversion	$O(n)$
Genetic Operation	$O(n)$
2-opt local exchange	$O(n^2)$

The time complexity is calculated for a single iteration. The decoding algorithm scans the n dimension particle for m number of time is $O(nm)$. Each particles are converted to positional value and vice versa is $O(n)$. The genetic operation crossover and mutation is performed for each particle is again $O(n)$, but the occurrence of this operation depends on the probability value. Similarly the local exchange heuristics consume $O(n^2)$, where it performs pair-wise swapping.

The NP-hard problem is now solvable with polynomial time because of its sub tasks.

VIII. CONCLUSION AND FUTURE WORK

This work applies genetic operator along with PSO to solve VRP. In the computational study the bench mark data instances are solved to evaluate the total route cost measure. This study enhances the solution quality of VRP. The space and time complexity of the proposed work is also analyzed, which shows it takes less time within polynomial time to solve VRP. The results obtained are competitive with benchmark results. The particles used are of n -dimension with varying inertia. Initial solutions are generated using greedy algorithm. Elitism is used to preserve the best obtained particles. GA operator's crossover and mutation are incorporated into PSO for better exploration of particles.

Further VRP can be extended to VRPTW, CVRPTW by including time constraint. The HPSO can be applied for MDVRP a variant of VRP by including more than one depot. This work can be extended for finding the route for garbage collection, school cabs, milk vans, etc. The work can also be applied for task assignment in multiprocessor environment by modifying the decoding procedure.

ACKNOWLEDGEMENT

The author wish to thank the anonymous reviewers for their comments to improve the paper quality.

REFERENCES

[1] Ai, T. J. and Kachitvichyanukul, V, "Particle swarm optimization for the capacitated vehicle routing problem", *International journal of logistics and SCM systems*, Vol 2, 2007, pp 50-55.
 [2] Ai, T. J. and Kachitvichyanukul, V, "A partice swarm optimization for the vehicle routing problem with simultaneous pickup and delivery", *Computers and operations research*, Vol 36, 2009, pp 1693-1702.

[3] Ai, T. J. and Kachitvichyanukul, V, "Particle swarm optimization and two solution representation for solving the capacitated vehicle routing problem", *Computer and industrial engineering*, Vol 56, 2009, pp 380-387.
 [4] Barrie M Baker, Ayeche M A, "A genetic algorithm for vehicle routing problem", *Computational and operational research*, Vol 30, 2003, pp 787-800.
 [5] Berger J and Barkaoui M, "A hybrid genetic algorithm for the capacitated vehicle routing problem", *Proceedings of genetic and evolutionary computation conference*, 2003, pp 646-656.
 [6] Bo Liua, LingWanga, Yi-Hui Jina, "An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers", *Computers & Operations Research*, Vol 35, 2008, pp 2791 – 2806.
 [7] Chen Ai-ling, Yang Gen-Ke, Wu zhi-ming, "Hybrid discrete paticle swarm optimization algorithm for capacitated vehicle routing problem", *Journal of Zhejiang University*, Vol 7(4), 2006, pp 607-614.
 [8] Dantzig G B and Ramset J H, "The truck dispatching problem," *Management science*, Vol 6, 1958, pp 81-91.
 [9] Fatih Tasgetiren, Angela Chen, Gunes Gencyilmaz and Said Gattoufi, "Smallest Position Value Approach", *Differential Evolution, Springer-Verlag*, 2009, pp 121-138.
 [10] Gendreu M, Hertz A, Laporte G, "A tabu search heuristic for the vehicle routing problem", *Management Science*, Vol 40, 1994, pp 1276 – 1290.
 [11] Guido perboli, Ferdinando Pezzella, Roberto Tadei, "EVE-OPT: A Hybrid algorithm for CVRP", *Mathematical Methods of Operations Research*, 2008, pp 361-382.
 [12] Hrvoje Marković, Ivana Čavar, Tonči Carić, "Using datamining to forecast uncertain demands in stochastic vehicle routing problem," *Proceedings of the 13th International Symposium on Electronics in Traffic – intelligent transport systems*, 2004.
 [13] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp 1942-1948.
 [14] Luo X. G. and Shi H. B, "Improved Particle Swarm Optimization for Vehicle routing problem with Non-full load", *Journal of East China University of Science and Technology(Natural Science Edition)*, Vol 32(7), 2006, pp 767-771.
 [15] Michalewicz Z, "Genetic Algorithms + Data Structures = Evolution Programs: A Hierarchy of Evolution Programs", *Berlin: Springer*, 1996.
 [16] Ning, L, Tong, Z and Debaio S, "Particle swarm optimization for vehicle routing problem", *Journals of system engineering*, Vol 19, 2004, pp 596- 600.
 [17] Osman I H, "Metastrategy simulated annealing and tabu search algorithm for vehicle routing problem," *Annual of operation research*, Vol 41, 1993, pp 421-451.
 [18] Shao, Z. J, Gao, S. P, Wang S. S, "A hybrid particle swarm optimization algorithm for vehicle routing problem with stochastic travel time", *Fuzzy information and engineering*, Vol 54, 2009, pp 566- 574.
 [19] Thangiah,S.R. and A.V.Gubbi,"Effect of genetic sectoring on vehicle routing problems with time windows", *IEEE International Conference on Developing and Managine Intelligent System Projects*, 1993, 146-153.
 [20] Wu, Y, Ye, C. M. and Ma, H. M, "Parallel particle swarm optimization algorithm for vehicle routing problem with time windows", *Computer Engineering and Applications*, Vol 43(14, 2007), pp 223-226.
 [21] Y.-L. Xu, M.-H. Lim and M.-J. Er, "Investigation on Genetic Representations for Vehicle Routing Problem", *in*

Systems, Man and Cybernetics, IEEE International Conference, 2005, pp. 3083-3088

[22] Zhang Y, Jin Jeng Liu, Fu Duan Jing Ren, "Genetic algorithm in vehicle routing problem", *Intelligent information hiding and multimedia signal processing*, Vol 2, 2007, pp 578-581.

[23] Zhang. N. Z. and Wu. Y. H, "Particle Swarm Optimization with near neighbourhood factor based on vehicle routing problem", *Computer Engineering and Applications*, Vol 44(32, 2008), pp 216-218.

S. Geetha, Completed her Masters in Computer Science during, 1997. Doing her Ph.D., for past five years in the area of soft computing techniques for optimization problem. Working as a Senior Lecturer in the Department of Mathematics and

Computer Applications for more than 13 years at PSG College of Technology, Coimbatore, Tamil Nadu, India.

G. Poonthalir, completed her Masters of Computer Applications during 2002. She was working as a Lecturer for more than 5 years. Presently working as Lecturer in the Mathematics and Computer Applications department at PSG College of Technology, Coimbatore, Tamil Nadu, India.

P. T. Vanathi, completed her doctorate in Electronics and Communication Engineering. She is having more than 80 paper publications. She guided more than 10 research scholars. She is working as Assistant Professor from 1987, in Electronics and Communication Engineering Department at PSG College of Technology, Coimbatore, Tamil Nadu, India.