

Meta Heuristic Algorithms for Vehicle Routing Problem with Stochastic Demands

Geetha Shanmugam, Poonthalir Ganesan and Dr. P.T. Vanathi
Department of Mathematics and Computer Applications,
PSG College of Technology, Tamil Nadu, India 641 004, India

Abstract: **Problem statement:** The shipment of goods from manufacturer to the consumer is a focal point of distribution logistics. In reality, the demand of consumers is not known a priori. This kind of distribution is dealt by Stochastic Vehicle Routing Problem (SVRP) which is a NP-hard problem. In this proposed work, VRP with stochastic demand is considered. A probability distribution is considered as a random variable for stochastic demand of a customer. **Approach:** In this study, VRPSD is resolved using Meta heuristic algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Hybrid PSO (HPSO). Dynamic Programming (DP) is used to find the expected cost of each route generated by GA, PSO and HPSO. **Results:** The objective is to minimize the total expected cost of a priori route. The fitness value of a priori route is calculated using DP. In proposed HPSO, the initial particles are generated based Nearest Neighbor Heuristic (NNH). Elitism is used in HPSO for updating the particles. The algorithm is implemented using MATLAB7.0 and tested with problems having different number of customers. The results obtained are competitive in terms of execution time and memory usage. **Conclusion:** The computational time is reduced as polynomial time as $O(nKQ)$ time and the memory required is $O(nQ)$. The ANOVA test is performed to compare the proposed HPSO with other heuristic algorithms.

Key words: Genetic algorithm, Particle Swarm Optimization (PSO), Dynamic Programming (DP), Vehicle Routing Problem (VRP), stochastic demands, hybrid PSO, distribution logistics, random variables, priori route, exact algorithm

INTRODUCTION

The VRP was first introduced by (Dantzig and Ramset, 1958), where the objective is to find the optimal set of routes for a fleet of vehicles in order to serve a given set of customers. Also a VRP in a broad sense is a generic name given to whole class of problems in which a set of routes for a fleet of vehicles based at one or several depots must be determined for a number of geographically dispersed cities or customers. Because of its overall presence in the fields of transportation, distribution and logistics VRP arises as one of the most challenging combinatorial optimization tasks even it is defined more than forty years ago. Figure 1 shows an example of VRP with 16 customers serviced by three vehicles from a central depot.

It is easy to describe the VRP problem, but difficult to solve. VRP usually takes exponential time for finding the optimal solution, is a NP-hard problem. There are many solutions are proposed to solve VRP but finding a globally minimum solution is computationally complex.

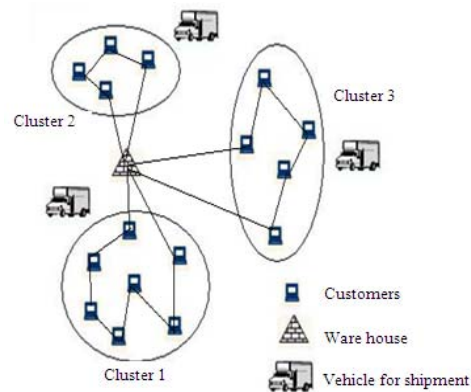


Fig. 1: Example of VRP

Many approaches are developed to solve VRP:
Exact algorithms: Branch and bound (Zhang *et al.*, 2010, Fisher, 1994) and branch and cut (Bektas *et al.*, 2009), **Heuristic algorithms:** (Battarra, 2010) and **Meta heuristic algorithm:** Genetic Algorithm (GA) (Sarabian and Lee, 2010; Nazif and Lee,

Corresponding Author: S. GEETHA, Department of Mathematics and Computer Applications, PSG College of Technology, Tamil Nadu, India 641 004

2010), Ant colony optimization (ACO) (Reimann *et al.*, 2004, Yu *et al.*, 2008) and Particle Swarm Optimization (PSO) (Kanthavel and Prasad, 2011; Shanmugam *et al.*, 2010; Srichandum and Rujirayanyong, 2010) are developed by many researchers for deterministic VRP. But they are not suitable for many real-time applications. Obviously, VRP with stochastic demands are most challenging and sophisticated than VRP. A VRPSD is an NP-hard problem, solving it by exact algorithm is time consuming and computationally intractable. There exist many real-time applications that motivated the research in the field of VRPSD, such as Courier services, Emergency services, Milk distribution, Taxi cab services and Refuse collection management. The delivery of petroleum products, home heating oil and industrial gases was already cited by (Chepuri and Homam-de-Mello, 2005; Psaraftis, 1985) VRPSD arises even in delivering mails, packages, items to hospitals and supplies to cities under emergency state. In a company during its strategic planning phase, designing a route before knowing the exact demands of a customer is cited by (Bertsimas, 1992) as a practical application of VRPSD.

The Stochastic VRP (SVRP) problems are new class of problems that allow information to be obtained and processed in real time. SVRP arise whenever some elements of the problem are random for example, stochastic demands (Erera *et al.*, 2010,) and stochastic travel times (Shao *et al.*, 2009, Xiangyong *et al.*, 2010). Only relatively small instances solved to optimality and good heuristics are hard to design and assess. The basic input data to solve such a problem are the probability density functions of the random variables representing demands at the nodes (Gendreau *et al.*, 2009; 1996). In order to verify the probability density function, demands at the nodes must be recorded over a long period of time and a detailed statistical analysis of collected data must be made. On the other hand, the demands at some nodes are not available.

Heuristic algorithm for VRPSD: The same recourse technique is followed Laporte *et al.* (2004), for solving VRPSD using tabu search and L-Shaped algorithm. VRPSD is also solved using Meta heuristics by (Bianchi *et al.*, 2004). Iterated Local search, Tabu search, Simulated Annealing, Ant colony optimization and Evolutionary algorithm are the five Meta heuristics algorithm implemented and found better solution quality in respect to cyclic heuristic.

The PSO is a stochastic optimization search algorithm. PSO is a population based search algorithm proposed by Kennedy and Eberhart (1995) Shao *et al.* (2009) they adopt a mixed approach GL-PSO for solving VRP with stochastic travel time discussed combining population search algorithm with the guided local search GLS based on trajectory search algorithm to solve SVRP. Yong and Hai-Ying, (2008) they used PSO with Inver-over operator and Dynamic Programming to solve VRPSD.

Genetic Algorithm (GA) is a stochastic optimization technique; it avoids getting trapped in a local optimum by the operator mutation. Many researches have been done in VRP using GA (Ismail and Irhamah, 2008). proposed a new scheme based on a hybrid GA and Tabu Search heuristic for VRPSD with preventive restocking. The Breeder Genetic Algorithm (BGA) is also proposed to solve VRPSD by (Irhamah and Ismail, 2009). BGA is powerful and reliable in global searching than GA.

In this study Genetic Algorithm as well as Particle swarm optimization (PSO) incorporated with Dynamic Programming (DP) to resolve SVRP. In order to find even better result PSO with GA operator is developed along with NNH for VRPSD. Cluster first and route second is the way of approach adapted in this study to target VRPSD.

MATERIALS AND METHODS

Mathematical formulation: VRPSD is defined on complete graph $G = (V, A, D)$, where $V = \{v_0, v_1, v_2, \dots, v_n\}$ is a set of nodes (customers) with node v_0 denoting the depot, $A = \{(v_i, v_j) : i, j \in \{0, 1, \dots, n\}, v_i, v_j \in V, v_i \neq v_j\}$ is the set of arcs joining the nodes and $D = \{d_{ij} : v_i, v_j \in V, v_i \neq v_j\}$ are the travel costs (distances) between nodes. One vehicle with capacity Q has to deliver goods to the customers according to their demands. The customers' demands are stochastic variables ζ_i , $i=1, \dots, n$ independently distributed with known distributions. The actual demand of each customer is only known when the vehicle arrives at the customer location. It is also assumed that ζ_i does not exceed the vehicle's capacity Q and follows a discrete probability distribution $p_{ik} = \text{Probability}(\zeta_i = k)$, $k = 0, 1, 2, \dots, K \leq Q$. A route must start at the depot visit number of customers and return to the depot. A feasible solution to VRPSD is a permutation of the customers $V = \{v_0, v_1, v_2, \dots, v_n\}$ starting and ending at depot is called as a priori route. The vehicle visits the customer in the order of a priori route and depending on the actual customers demand besides whether to proceed to next customer or go to depot for restocking. Restocking occurs if the total demand of a customer exceeds the vehicle capacity.

In VRPSD the objective is to find an a priori route that minimizes the expected distance travelled by a vehicle based at depot with the capacity Q . VRPSD under restocking policy requires to find a vehicle route and restocking policy to decide whether or not, to return to the depot for restocking before visiting the next customer to minimize the total expected cost of travel. Sometimes the choice of restocking is the best one, even if the vehicle is not empty, or if its capacity is bigger than the expected demand of the next scheduled customer, this action is called ‘preventive restocking’. The total cost depends on:

- Cost of travel from one customer to another customer
- Cost of restocking (travelling cost from customer to depot)
- Cost of travel from customer to depot for restocking, if demand of customer exceeds the vehicle capacity

Suppose that $V = \{v_0, v_1, v_2, \dots, v_n\}$ is a priori route. After distribution, let the remaining load of vehicle be q and $f_i(q)$ is the expected route length of the vehicle after servicing customer j , so that the expected cost of a priori route is $f_0(Q)$. If H_j denotes the set of all possible loads that a vehicle can have after service completion at customer j . Then $f_i(q)$ for $q \in H_j$ satisfies:

$$f_i(q) = \text{Minimum} \{f_j^p(q), f_j^r(q)\} \tag{1}$$

where:

$$f_j^r(q) = d_{j,0} + d_{0,j+1} + \sum_{k=1}^K f_{j+1}(Q-k)p_{j+1,k} \tag{2}$$

$$f_j^p(q) = d_{j,j+1} + \sum_{k:k \leq q} f_{j+1}(q-k)p_{j+1,k} + \sum_{k:k \geq q} [2d_{j+1,0} + f_{j+1}(q+Q-k)]p_{j+1,k} \tag{3}$$

With the boundary condition $f_n(q) = d_{n,0}, q \in H_n$.

In (2), $f_j^r(q)$ is the expected cost corresponding to the choice of proceeding directly to the next customer, while (3), $f_j^p(q)$ is the expected cost in case preventive restocking is chosen. These equations are used recursively to determine the

objective value of the planned vehicle route and the optimal sequence of decisions after customers are served (Yong and Hai-Ying, 2008). This study focuses on designing PSO incorporated with GA operators such as mutation and crossover to solve VRPSD and DP is used to solve the objective function of stochastic program.

Proposed work: In this study, GA, PSO and HPSO are the Meta heuristics approaches adapted to solve VRPSD in this study. Solutions are represented using Permutation encoding in both GA and PSO. First GA with DP is developed to solve VRPSD. Then PSO with permutation encoding is proposed as second approach. To have better exploration of the solution, PSO is combined with GA operators called as Hybrid PSO (HPSO) is proposed as a third work. The genetic operators always explore all possible solutions. The velocity of PSO is updated; to exploits the best obtained solution. In HPSO, the n dimension particles are used to represent the solutions. The solution is converted to particle position value and smallest position value is used for converting particle position back to solution.

Genetic algorithm: Genetic Algorithm is a stochastic optimization technique; it avoids getting trapped in a local optimum by the operator mutation. GA starts with an initial set of random solution called as population. Each solution in the population is called as chromosome. These chromosomes evolve through the iteration called as generations. During each generation the chromosomes are evaluated to fitness. The best fitting chromosomes are having high probability in getting selected for genetic operation such as crossover and mutation. After the predetermined generation, GA converges to the near optimal solution of the problem.

The algorithm to solve VRPSD using GA is shown in the Fig. 2. The feasible solutions are generated using NNH so that the length of route is less and help to converge quickly. The distance matrix is calculated among all pairs of customers and between a depot and all customers using Euclidean distance metric.

Initialization and improvement: The real coded chromosomes are used to represent the solutions. For example, 2 3 4 1 5 is one of the solution represented as the chromosome for five customers. Then based on their demands p_{ik} and remaining load in the vehicle, vehicle either go to the next customer or back to the depot to load the vehicle.

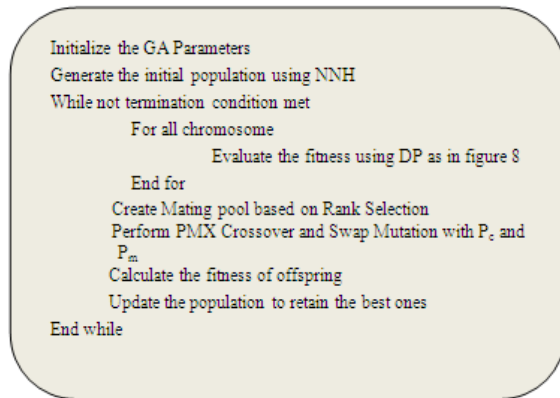


Fig. 2: GA Algorithm for VRPSD

The feasible solutions are generated using NNH. The NNH is used to generate the path in such a way that the customers nearby are grouped in the same route. The distance matrix is calculated among all pairs of customers and between each depot and all customers using Euclidean distance metric. Initially, half of the populations are generated using NNH. Remaining chromosomes are generated randomly.

Evaluation: The fitness is the route cost $f_j(q)$ of a vehicle is calculated using dynamic. The fitness is high, if the total cost is minimum.

Selection: The rank selection is used for choosing the chromosome to undergo genetic operations. The chromosomes are arranged in non decreasing order based on their fitness value. Then the chromosomes are selected for mating pool with high fitness and low fitness equally.

Genetic operations: The two essential needs of searching such as exploration and exploitation are achieved by mutation and crossover operation of GA. The PMX crossover is used since the chromosomes are represented as path representation. Swap and inversion mutation is used for mutate operation.

Particle swarm optimization: In PSO, the potential solutions are called as particles in the search space. All particles have fitness values evaluated using fitness function and velocities directing the particles towards solution. The PSO is used to solve VRPSD with the real coded particles similar to that of chromosomes represented in GA. The algorithm is similar to general PSO except the particle initialization and conversion.

Particle initialization and conversion: Initial solutions of PSO are initialized like chromosomes of GA. Every solution represents the vehicle route of all depots. It should be converted to particle position value (Liua *et al.*, 2008) using the following equation:

$$x_{ij} = x_{\min} + (x_{\max} - x_{\min}) / n [(y_{ij} - 1 + \text{rand}())] \quad (4)$$

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 (4) is shown in the following Table 3.

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 Rank Order Value (ROV). The ROV uses Smallest Position Value (SPV) [34, 35] 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 Table 4 illustrates the ROV of a particle to a permutation encoding. The pseudo code for solving VRPSD using PSO is shown in the Fig. 3.

Hybrid particle swarm optimization for VRPSD: The PSO has many limitations. The performance of PSO is more problem dependent. The parameters of PSO vary from one problem to another, finding the optimal parameter value is not an easy task. PSO may converge prematurely, because of the fast rate of information flow between particles to converge to a single point lies between the global best and the personal best positions, resulting in similar particles without diversity of solution search space. PSO is hybrid with GA to overcome its limitations. This hybrid PSO is expected to have merits of both PSO and GA. PSO are simpler in algorithm and able to control its convergence, when compared to GA. But to have diverse search space, the GA operators are used. It makes the particle to fly to new search space and avoid premature convergence. So HPSO is proposed to solve VRPSD with varying inertia. Elitism is also added in order to preserve the good particles without being updated by inertia and GA operators. The pseudo code of HPSO is shown in Fig 4.

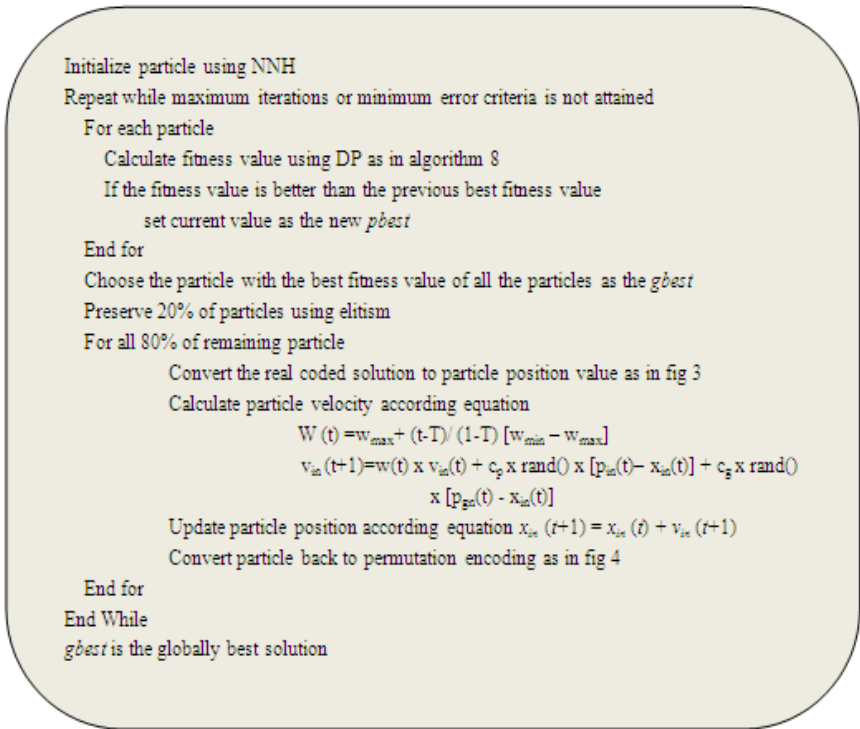


Fig. 3: PSO algorithm for VRPSD

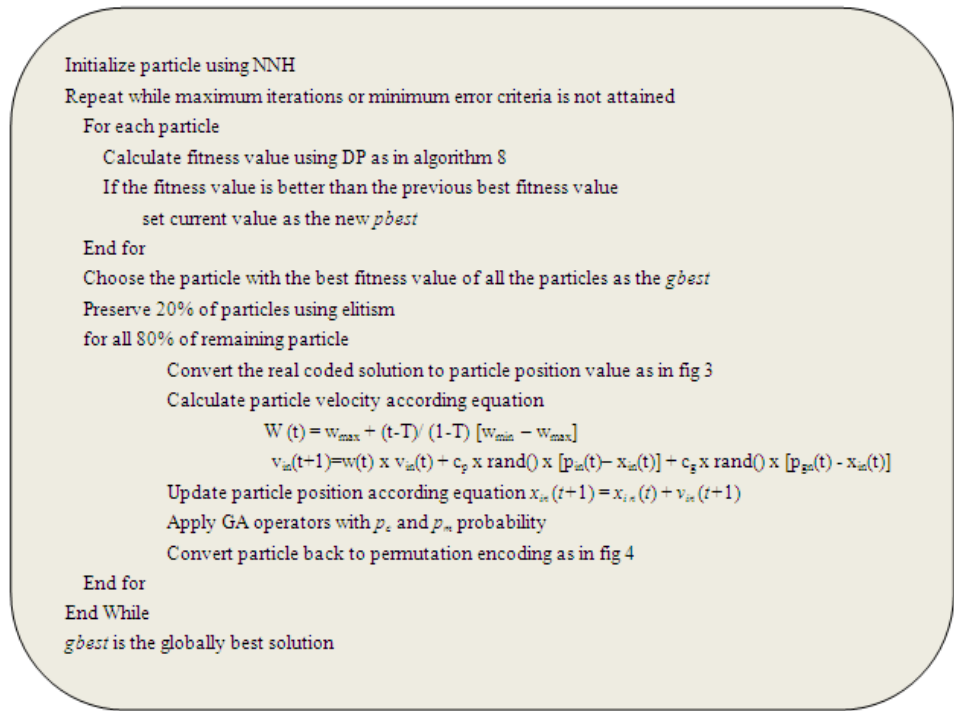


Fig. 4: HPSO Algorithm for VRPSD

Genetic operators: The crossover and mutation are the two operators applied for particles to evolve better in next iteration. Single point cross over is applied with all particles. The crossover is done with the gbest, (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 . If it is greater than P_m , inversion mutation is applied to explore the solution search space. The crossover and mutation is elaborated in Fig. 5.

```

For each particle  $X_i$ 
  Select a random point  $p$  between 1 to  $n$ 
  Swap the particles value with  $P_g$ , i.e.  $X_{ip}, X_{ip+1}, \dots, X_{id}$  with
   $P_{gp}, P_{gp+1}, \dots, P_{gd}$ 
  If  $\text{rand}() < P_m$ , then
    Two random points is selected  $p_1, p_2$  to perform swap
    mutation i.e. to swap  $X_{ip_1}$  and  $X_{ip_2}$ 
  Else
    Inversion mutation is done by swapping all the
    position from  $X_{ip_1}$  to  $X_{ip_2}$ 
  End If
End For
    
```

Fig. 5: Pseudo code for Genetic operators

Dynamic programming to evaluate the route cost:

The dynamic programming uses backward recursion that is more efficient than forward recursion. In VRPSD all possible combination of customer demands and vehicle load are to be explored. It consumes more space and time i.e. $O((k+1)^n)$. Recursion does not store the intermediate results. But DP consumes less space and time with caching strategy. In general DP model has three basic elements: stages, alternatives at each stage and state for each stage. In our proposed method, the three models of DP are:

- Visiting sequence of customer is known a priori, where each customer is considered as a stage
- Possible Load of vehicle H_j at each stage is considered as alternatives
- Possible demand of a customer ζ_i at each stage is called as state

Then a priori route is now evaluated to find the cost using DP is shown in Fig 6.

```

For each solution
  Initialize  $f_n(q) = d_{n,0}$ ,  $q \in H_{0n}$ 
  For  $j = n-1, n-2, \dots, 1$ 
    For  $q = 0, 1, \dots, Q$ 
      For  $k = 0, 2, \dots, K$ 
        If  $k \leq q$  Calculate
           $f_j^p(q) = f_{j+1}^p(q) + f_{j+1}(q-k)p_{j+1,k}$ 
        Else Calculate
           $f_j^p(q) = f_{j+1}^p(q) + [2d_{j+1,0} + f_{j+1}(q+Q-k)]p_{j+1,k}$ 
        End If
      End For
    End For
    Calculate
       $f_j^p(q) = f_j^p(q) + d_{j,j+1}$ 
    For  $k = 1, 2, \dots, K$  Calculate
       $f_j^r(q) = f_{j+1}^r(q) + f_{j+1}(Q-k)p_{j+1,k}$ 
    End For
    Calculate
       $f_j^r(q) = f_j^r(q) + d_{j,0} + d_{0,j+1}$ 
    Set  $f_j(q) = \text{minimum} \{f_j^p(q), f_j^r(q)\}$ 
    For  $k = 0, 1, \dots, K$  calculate  $f_0(Q) = f_0(Q) - f_1(Q-k)$  End For
  End For
End For
End
    
```

Fig. 6: Pseudo code for DP

Table 1: Conversion of solution to a particle

Dimension j	1	2	3	4	5
Permutation					
Encoding (Yi)	1	2	3	4	5
Position value (Xi)	0.304	0.582	0.807	1.528	1.777

Table 2: Particle Conversion to permutation encoding

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

Table 3: Results of VRPSD using GA

No. of customers	Initial best solution	Final best solution	No. of iteration	Improvement rate (%)
5	18.5829	18.5829	1	0
10	24.8939	22.3212	15	11.52
12	26.2050	25.1820	5	17.9

RESULTS

The Proposed algorithm is tested with the random data set. The data set is generated based on data provided Yong and Hai-Ying (2008). The capacity of the vehicle is considered as $Q = 15$. The number of customers/nodes to be serviced is considered for both small as well as medium size. The distance between the customers and the depot is generated randomly. The demand of each customer will not be known until the day of service, where it is assumed to lie between 0 and 6. Based on the previous data, the demand of each customer is given with discrete probability for demand varying from 0 to 6.

The Proposed Meta heuristic algorithms are implemented in MatLab 7.0.1 with Pentium 4.0, 2.3GHz. The parameters of PSO are Number of Particles $I = 10$, Number of Iteration $T=5$, Initial inertia weight $w_{min}= 0.4$, Last inertia weight $w_{max}=0.1$, personal best position acceleration constant $c_p=0.5$, Global best position acceleration constant $c_g=0.5$, local best position acceleration constant $c_l=1.5$ with the neighbor constant $K=5$. 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 $x_{min}=0.1$ and $x_{max}=0.7$ are values used for converting solution to particle.

The HPSO is tested against PSO and GA. The PSO algorithm is implemented with the above mentioned values for parameters. The PSO uses varying inertia with gbest, pbest and lbest. The NNH approach is used to generate the initial population of PSO. In this, crossover and mutation are not incorporated.

The GA is implemented with partially mapped crossover (PMX) and swap and inversion mutation. The parameters used are Population Size is 10,

Table 4: Results of VRPSD using PSO

No. of Customers	Initial best solution	Final best solution	No. of iteration	Improvement rate (%)
5	18.5829	18.5829	1	0
10	24.8939	23.1462	15	0.7
12	26.2050	25.7181	13	1.84

Table 5: Results of VRPSD using HPSO

No. of Customers	Initial best solution	Final best solution	No. of iteration	Improvement rate (%)
5	18.5829	18.5829	1	0
10	24.8939	21.9753	12	11.72
12	26.2050	24.6812	10	20.86

Number of generation is 20, crossover probability rate p_c is 0.6 and mutation probability rate p_m is 0.4 for either performing swap or inversion mutation. After testing the data set with different elitism percentages, it is confirmed to be 20% which provides better results in preserving the good particles.

The results obtained by the GA are depicted in the Table 3. It projects the results for various customer numbers such as 5, 10 and 12 with initial best solution, final best solution along with how fast it converge to provide the optimal result and also the improvement rate. The improvement rate is calculated as:

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

For the customer size 5, the improvement rate is 0. Because the initial chromosomes are generated using NNH, which itself produced the best solution. The number of iteration also increases in order to produce the optimal solution, when the number of customers is 10 and 12.

The results of PSO are projected same as GA in the Table 4. The improvement rate of PSO is low when compared to GA. Since NNH is used the initial best solution is same in both the cases. Table 5 shows the results obtained using HPSO. The improvement rates are higher when compared to GA as well as PSO.

DISCUSSION

The performance of Meta heuristic algorithms for VRPSD is shown in the Fig. 7. The graph depicts the cost obtained for the vehicle capacity $Q = 10$. The x-axis refers customer size and y-axis refers the cost of a priori route.

To investigate whether the performance of the algorithm are different with respect to problem size,

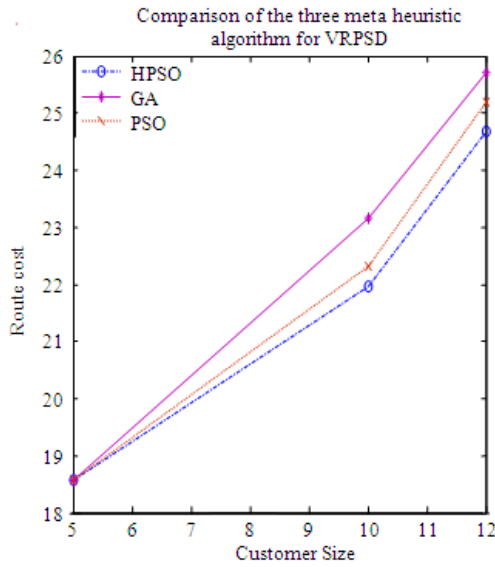


Fig. 7: Comparison of Meta heuristic algorithm for VRPSD

ANOVA test was performed. There is no prominent difference at 5% level of significance in their performance. Another ANOVA test was performed with respect to number of iterations for the three algorithms. Table 5 shows the results obtained by ANOVA. Since the value obtained for F is less than the value of $F_{0.05}$ with 2 and 8 degrees of freedom the null hypothesis can be accepted at 5% level of significance. The performance of algorithms differs with the number of iterations.

The cost of a priori route using HPSO without NNH and elitism for 10 random customers is shown in the Fig. 8. In iteration 2, the cost of route is reduced from 23.9 to 22.5, but this less cost is not preserved for further iteration, so after 15th iteration it converges to optimal solution.

In order to preserve the best particles elitism are added to HPSO. Because of which the convergence are made faster when compared to the one shown in Fig. 9-10 shows the performance of HPSO with elitism and without NNH.

Because of random initialization the initial cost is high. In order to reduce the initial cost, heuristic NNH is added to HPSO along with elitism. The Fig. 10 shows the route cost obtained by HPSO with elitism and NNH. And as given in the Table 6 the improvement rate is also higher.

Computational complexity: As stated, the complex NP-hard problem VRPSD is decomposed into sub

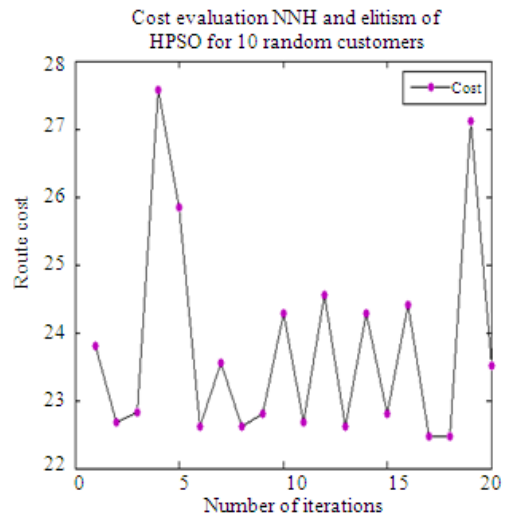


Fig. 8: Route cost of HPSO without NNH and elitism

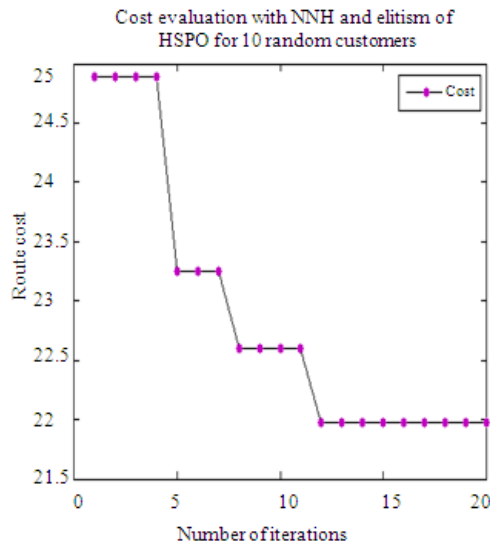


Fig. 9: Route cost of HPSO without NNH and with elitism

problems which are relatively solvable. Let n is the number of customers and K is the quantity that a vehicle can have at an instance of time and Q is the maximum quantity a vehicle can hold. In evaluating the fitness of each solution(DP) the computation of $f_0(q)$ runs in $O(nKQ)$ time, the memory required is $O(nQ)$, if all intermediate values of $f_j(q)$ are stored, otherwise $O(Q)$. This calculation has to be done for all solutions of PSO, HPSO and GA.

Table 6: ANOVA table

Source of Variation	Degrees of freedom	Sum of squares	Mean Square	F
Algorithms	2	11.555666	5.777833	1.88940
Error	8	87.3333	10.916662	
Total	10	98.8889		

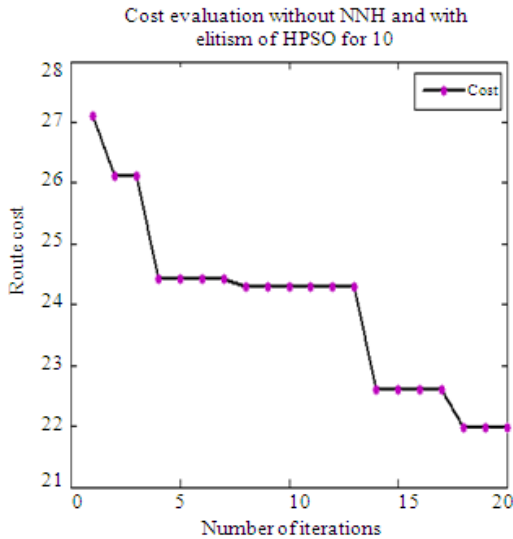


Fig. 10: Route cost of HPSO with NNH and with elitism

CONCLUSION

This study applies genetic operator along with PSO to solve VRPSD. This study enhances the solution quality of VRPSD. The results obtained are competitive. The particles used are of n-dimension for n customers. It uses varying inertia. Elitism is used to update the particle to preserve the best obtained particles. To explore the results crossover and mutation are the operators added to PSO. To start with the good quality solution, NNH is used to generate the initial solution. The complexity of the proposed algorithm is also in polynomial time. The particles of PSO are in permutation encoding, which can be used for solving many sequencing problems like Traveling Salesman Problem, Bin Packing etc.

Further VRPSD can be extended to VRP with stochastic time. It can also be used for waste collection vehicle routing problem and for bus transportation where the demands are stochastic.

REFERENCES

Battarra, M., 2010. Exact and heuristic algorithms for routing problems. 4OR: A Quarterly J. Operat. Res. DOI: 10.1007/s10288-010-0141-9

Bektas, T., G. Erdogan, S. Ropke, 2009 Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. *Trans. Sci.*, DOI: 10.1287/trsc.1100.0352

Bertsimas, D.J., 1992. A vehicle routing problem with stochastic demand. *Operat. Res.*, 40: 574-585. DOI: 10.1287/opre.40.3.574

Bianchi, L., M. Birattari, M. Chiarandini, M. Manfrin and M. Mastrolilli *et al.*, 2004. Metaheuristics for the vehicle routing problem with stochastic demands. *Parallel Prob. Solv. Nature-PPSN VIII*, 3242: 450-460. DOI: 10.1007/978-3-540-30217-9_46

Chepuri, K. and T. Homam-de-Mello, 2005. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annal. Operat. Res.*, 134: 53-181. DOI: 10.1007/s10479-005-5729-7

Dantzig, G.B. and J.H. Ramset, 1958. The truck dispatching problem. *Manage. Sci.*, 6: 80-91.

Erera, A.L., J.C. Morales and M. Savelsbergh, 2010. The vehicle routing problem with stochastic demand and duration constraints. *Trans. Sci.*, 44: 474-492. DOI: 10.1287/trsc.1100.0324

Fisher, M.L., 1994. Optimal solution of vehicle routing problems using minimum K-trees. *Operat. Res.*, 42: 626-642.

Gendreau, M., G. Laporte and R. Seguin, 1996. Stochastic vehicle routing. *Eur. J. Operat. Res.*, 88: 3-12.

Gendreau, M., G. Laporte and R. Seguin, 2009. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operat. Res.*, 44: 469-477. DOI: 10.1287/opre.44.3.469

Irhamah, I. and Z. Ismail, 2009. A breeder genetic algorithm for vehicle routing problem with stochastic demands. *J. Applied Sci. Res.*, 5: 1998-2005.

Ismail, Z. and Irhamah, 2008. Solving the vehicle routing problem with stochastic demands via hybrid genetic algorithm-tabu search. *J. Math. Statis.*, 4: 161-167. DOI: 10.3844/jmssp.2008.161.167

Kanthavel, K. and P. Prasad, 2011. Optimization of capacitated vehicle routing problem by nested particle swarm optimization. *Am. J. Applied Sci.*, 8: 107-112.

Liua, B., L. Wanga and Y.H. Jina, 2008. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Comput. Operat. Res.*, 35: 2791-2806. DOI: 10.1016/j.cor.2006.12.013

- Nazif, H. and L.S. Lee, 2010. Optimized crossover genetic algorithm for vehicle routing problem with time windows. *Am. J. Applied Sci.*, 7: 95-101. DOI: 10.3844/ajassp.2010.95.101
- Psaraftis, H.N., 1985. Dynamic vehicle routing: Status and prospects. *Ann. Operat. Res.*, 61: 143-164. DOI: 10.1007/BF02098286
- Reimann, M., K. Doerner and R.F. Hartl, 2004. D-ants: Savings based ants divide and conquer the vehicle routing problem. *Comput. Operat. Res.*, 31: 563-591. DOI: 10.1016/S0305-0548(03)00014-5
- Sarabian, M. and L.V. Lee, 2010. A modified partially mapped multicrossover genetic algorithm for two-dimensional bin packing problem. *J. Math. Stat.*, 6: 157-162. DOI: 10.3844/jmssp.2010.157.162
- Shanmugam, G., P. Ganesan and P.T. Vanathi, 2010. A hybrid particle swarm optimization with genetic operator for vehicle routing problem. *J. Adv. Inform. Technol.*, 1: 181-188. DOI: 10.4304/jait.1.4.181-188
- Shao, Z.J. S.P. Gao and S.S. Wang, 2009. A hybrid particle swarm optimization algorithm for vehicle routing problem with stochastic travel time. *Fuzzy Infor. Eng.*, 54: 566-574. DOI: 10.1007/978-3-540-88914-4_70
- Srichandum, S. and T. Rujiranyong, 2010. Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization. *Am. J. Eng. Applied Sci.*, 3: 7-14. DOI: 10.3844/ajeassp.2010.7.14
- Xiangyong, L., P. Tian and S.C.H. Leung, 2010. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *Int. J. Prod. Econ.*, 125: 137-145.
- Yong, P. and Z. Hai-Ying, 2008. Research on vehicle routing problem with stochastic demand and PSO-DP algorithm with inver-over operator. *Syst. Eng. Theory Practice*, 28: 76-81. DOI: 10.1016/S1874-8651(10)60003-8
- Yu, B., Z.Z. Yang, B. Yao, 2008. An improved ant colony optimization for vehicle routing problem. *European J. Operat. Res.*, 196: 171-176. DOI: 10.1016/j.ejor.2008.02.028
- Zhang, Z., H. Qin, A. Lim and S. Guo, 2010. Branch and bound algorithm for a single vehicle routing problem with toll-by-weight scheme. *Trends Applied Intell. Syst.*, 6098: 179-188. DOI: 10.1007/978-3-642-13033-5_19