Nested particle swarm optimisation for multi-depot vehicle routing problem

S. Geetha* and G. Poonthalir

Department of Mathematics and Computer Applications,
PSG College of Technology,
Coimbatore 641004, Tamilnadu, India
E-mail: geet_shan@yahoo.com
E-mail: thalirkathir@rediffmail.com
*Corresponding author

P.T. Vanathi

Department of Electronics and Communication Engineering,
PSG College of Technology,
Coimbatore 641004, Tamilnadu, India
E-mail: ptvani@yahoo.com

Abstract: Vehicle routing problem (VRP) is a well-known non-deterministic polynomial hard problem in operations research. VRP is more suited for applications having one warehouse. A variant of VRP called as multi-depot vehicle routing problem (MDVRP) has more than one warehouse. Cluster first and route second is the methodology used for solving MDVRP. An improved *k*-means algorithm is proposed for clustering that reduces the MDVRP to multiple VRP. In this work, MDVRP is considered with more than one objective and nested particle swarm optimisation with genetic operators is proposed for solving each VRP. Master particle swarm optimisation forms the group within each cluster. Slave particle swarm optimisation generates the route for each group. The objective of MDVRP is to minimise the total travel length along with route and load balance among the depots and vehicles. The results obtained are better in balancing load, route length and the number of vehicles, rather than minimisation of total cost.

Keywords: improved *k*-means algorithm; NPSO; nested particle swarm optimisation; genetic operator; local exchange; MDVRP; multi-depot vehicle routing problem; home delivery pharmacy programme; waste collection management.

Reference to this paper should be made as follows: Geetha, S., Poonthalir, G. and Vanathi, P.T. (2013) 'Nested particle swarm optimisation for multi-depot vehicle routing problem', *Int. J. Operational Research*, Vol. 16, No. 3, pp.329–348.

Biographical notes: S. Geetha completed her Masters in Computer Science in 1997. She is doing her research for past six years in the area of soft computing techniques for optimisation problem. She is working as an Assistant Professor (Senior Grade) in the Department of Mathematics and Computer Applications at PSG College of Technology, Coimbatore, Tamilnadu, India, for 14 years. She has three international papers published in the area of PSO for VRP.

G. Poonthalir completed her Masters of Computer Applications in 2001. She is working as an Assistant Professor in the Department of Mathematics and Computer Applications at PSG College of Technology, Coimbatore, Tamilnadu, India. She is doing her research for the past three years in the area of soft computing.

P.T. Vanathi completed her Doctorate in Electronics and Communication Engineering. She is having more than 80 publications at national and international levels. She has produced more than ten PhD. She is working as an Associate Professor, from 1987, in the Department of Electronics and Communication Engineering at PSG College of Technology, Coimbatore, Tamilnadu, India.

1 Introduction

Vehicle routing problem (VRP) was first introduced five decades before by Dantzig and Ramset (1958) is an important technique in solving logistic distribution management. It is extensively studied because of its wide application in both public and private sectors. Often, the context of VRP is to deliver the goods located at central depot to customers/cities that have placed order for goods. Many approaches are developed to solve single-depot VRP. But single-depot VRP is not suitable for many real-time applications. Obviously, VRP with multiple depots are most challenging and sophisticated than single-depot VRP. Multi-depot vehicle routing problem (MDVRP) is an important and challenging problem in logistics management. A MDVRP is an non-deterministic polynomial-hard (NP-hard) combinatorial optimisation problem, solving it by exact algorithm is time consuming and computationally intractable. MDVRP is a generalisation of single-depot VRP in which vehicle(s) start from multiple depot and return to their depot of origin. The traditional objective of MDVRP is to minimise the tour length, this can be handled with additional constraints and assumptions. There exist many real-time applications that motivated the research in the field of MDVRP, such as courier services, emergency services, newspaper distribution, taxi cab services and refuse collection management.

Particle swarm optimisation (PSO) is a population-based evolutionary computation technique developed by Eberhart and Kennedy (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995), inspired by social behaviour of bird flocking or fish schooling. PSO is a computational intelligence-based technique that is not largely affected by the size and can be easily implemented on computer and converge to the optimal solution in many problems. PSO can solve a variety of difficult optimisation problems. It has a flexible and well-balanced mechanism to enhance and adapt to the global and local exploration abilities. Its simplicity in coding and consistency in performance motivated the usage of PSO for VRP. Pongchairerks and Kachitvichyanukul (2009) used PSO for solving job shop scheduling problem. Ai and Kachitvichyanukul (2009a,b) proposed a solution for VRP with time window using PSO.

This paper focuses in solving MDVRP by nested particle swarm optimisation (NPSO) algorithm to find optimal solution efficiently and effectively. The initial clustering of customers is done by the *k*-means algorithm. An additional constraint of priority is included with *k*-means clustering the customers to the nearest depot using minimax principle. First-fit decreasing algorithm is used for allocating customers to the depot, i.e. based on their demands. NPSO consists of master PSO and slave PSO. Master PSO is applied for assigning the customers to the vehicle within the depot/cluster. Slave PSO is used for forming the routes within each group of the cluster formed by master PSO. To avoid trapping into local optima, the genetic operators' mutation and crossover are used in NPSO. The nearest neighbour heuristic (NNH) and 2-opt local exchange are incorporated into slave PSO to generate better route in short span of time. The main objective of this paper is in balancing the load and route of each depot/vehicle, in addition to minimising the total route cost.

2 Literature survey

Even though there are number of research projects for VRP with single depot, it can be rarely solved to optimal solution only when the customers are few hundred. MDVRP is a problem in which finding an optimal solution is impossible even for small size problem instances. Comparatively, the number of research projects on the MDVRP is fewer. In literature, there are many published work dealing with the traditional MDVRP. The first heuristics were proposed by Tillman (1969). Gillet and Johnson (1976) used a clustering procedure and a sweep heuristic for each depot. Raft (1982) proposed a multi-phase approach with additional refinement. Ball et al. (1983) used route first and cluster second approach. Linear programming and heuristics were combined by Klots et al. (1992). Chao et al. (1993) used a simple initialisation heuristic and improvement method. Potvin and Rousseau (1993) suggested few ideas for assigning the customers to the depot. Renaud et al. (1996) proposed a new heuristic for solving MDVRP using tabu search with route and capacity restriction. Su (1999) proposed a method for routing a vehicle based on location, quantity and due date of demand at real time. Giosa et al. (2002) proposed a general two-stage approach called 'assignment first and route second' by comparing the assignment of customers to depot using six different heuristics for MDVRP with time window. Mingozzi (2005) proposed an integer programming formulation and an exact method for solving periodic VRP and MDVRP. Lim and Wang (2005) introduced MDVRP with fixed distribution of vehicles. In that they proposed two methods one-stage and two-stage approaches. A combined heuristic algorithm for the MDVRP with inter-depot routes was proposed by Zhen and Zhang (2009). In Sombuntham and Kachitvichayanukul (2010), improved PSO is used for solving MDVRP with mutation and improved inertia. MDVRP with pickup and delivery is solved using PSO with multiple social learning structures in Wenjing and Ye (2010). Nilay and Nihan (2011) proposed a new type of geometric shape-based genetic clustering algorithm for solving MDVRP. MDVRP is solved using parallel ant colony optimisation by Yu et al. (2011) in which a virtual central depot is added for the MDVRP that becomes a similar problem of VRP with virtual central depot as the origin. In Zhang et al. (2011), scatter search (SS) for MDVRP with weight-related cost considers MDVRP by including freight cost. In that the sweep algorithm and the optimal splitting procedure are used to construct the initial trial solutions. An iterative descending algorithm and an arc choosing method are adopted in the SS to improve and combine the solutions, respectively.

From the study of literature, it is clear that the MDVRP is solved by generating the initial feasible solution and then improving and refining the solution for better results.

Due to its complexity, finding the optimal solution is time consuming for MDVRP. PSO proves to work faster than genetic algorithm (GA) from its literature to solve VRP. Many researchers applied PSO for VRP. This motivated to apply PSO for solving MDVRP. In this paper, NPSO is proposed, in which GA operators are used with slave PSO to provide better results (Geetha et al., 2010). The PSO uses varying inertia as stated by Ai and Kachitvichyanukul (2007, 2009a,b). To provide better solution quickly, the population are initialised using NNH. Clustering algorithms are also used as a pre-processor to generate the initial population of best quality. The master PSO is used for assigning customers to the vehicles within the cluster and slave PSO takes care of forming the routes of each vehicle. The routes formed for each vehicle are improved using local search 2-opt.

3 Mathematical formulation

MDVRP's mathematical formulation is based on cluster first and route second methodology. Let $G = (V, E)$ be a graph, where V is the set of vertices partitioned into two subsets V_c and V_d, where $V_c = \{V_1, ..., V_n\}$ represents the city or customer set and $V_d = \{V_{n+1}, ..., V_{n+m}\}$ represents depots. E is the set of edges connecting the vertices. A cost matrix $C = (c_{ij})$ corresponds to distance or travel time. Each city v_i is associated with a non-negative demand q_i. The problems of interest are restricted to symmetric C and satisfies the triangle inequality, i.e. $c_{ij} = c_{ji}$ for all i, j and $c_{ik} \leq c_{ij} + c_{jk}$ for all i, j, k. At each depot, $V_{n+m} \in V_d$ is based on k identical vehicles of capacity Q. The MDVRP consists of constructing a set of vehicle routes in such a way that:

- Each route starts and ends at the same depot.
- Each customer is visited exactly once by a vehicle.
- The total demand of each route does not exceed the vehicle capacity Q.
- The total route cost is minimised.

As a combinatorial optimisation problem, MDVRP is now defined with objectives and constraints as follows:

Objectives

- Minimise the distance travelled 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.

In addition to the above specified constraints to have uniform distribution of load and work between the vehicles and the drivers which should also be considered to bring fairness into play. To achieve this, additional constraints are added to MDVRP. These constraints are taken care by first clustering the customers to the depot by means of packing them tightly to utilise the depot capacity.

- Load of all vehicles are balanced.
- Route cost is balanced among the depots.

The problem is given with a set of

Customers: $c_1, c_2, c_3, \ldots, c_n$

Depots: $d_1, d_2, d_3, \ldots, d_m$

Demands: $q_1, q_2, q_3, \ldots, q_n$

Vehicles at each depot: $v_1, v_2, v_3, \ldots, v_m$

Capacity: Q

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 (q_i) and capacity (Q) and number of vehicles at each depot are positive integers.

The formulation of MDVRP problem is based on the cluster formation and then routing within each cluster. The customers are first assigned to the nearest depot to form m clusters. Then, within each cluster, the vehicle routes are formed so that each customer is serviced exactly by one vehicle of a depot. The explanation of problem is as follows.

The n customers are grouped to form m clusters, with the constraint of their demand and location (x, y). The set C is partitioned into m subsets C_i. Then, $C_i \neq \phi$ for $i = 1, ..., m$, $C_i \cap C_j = \phi$ for $i = 1, ..., m, j = 1, ..., m$ and $i \neq j$ and $\bigcup_{I=1}^{m} C_i = C$.

A customer c_i is included to a subset only if the summation of customer demands in that subset is less than or equal to the capacity of the vehicle. The number of customers in each cluster is denoted by $n_1, n_2, ..., n_m$, such that $\sum_{j=1}^{m} n_j = n$. After clustering, each depot will have n_j number of customers assigned to it. The customers are again formed into k_j group, each group is for a vehicle based at corresponding depot. The number of vehicles based at a depot is k. The groups are formed based on the following condition that $\sum_{p=1}^{k_j} n_{jp} = n_j \quad \forall j = 1$ to m.

The variables X and Y represent the decision variables, where

$$x_{ijkm} = \begin{cases} 1 \text{ if vehicle } k \text{ in a depot } m \text{ moves from customer } i \text{ to customer } j \\ 0 \text{ otherwise} \end{cases}$$

$$y_{ikm} = \begin{cases} 1 \text{ if vehicle } k \text{ serves customer } i \text{ assingned to depot } m \\ 0 \text{ otherwise} \end{cases}$$

The mathematical formulation for MDVRP based on Mao-Xiang (2006) is described as follows. The objective is to find X which minimises,

$$Z = \sum_{p=1}^{m} \sum_{q=1}^{k_p} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ijqp} \tag{1}$$

s.t.

$$\sum_{i=1}^{n} q_i y_{iqp} \le Q \tag{2}$$

$$0 \le n_{jq} \le n_j \tag{3}$$

$$\sum_{q=1}^{k_j} n_{jq} = n_j \quad \forall j = 1 \text{ to } m \tag{4}$$

$$\sum_{j=1}^{m} n_j = n \tag{5}$$

$$\sum_{p=1}^{m} \sum_{q=1}^{k_p} y_{iqp} = 1 \tag{6}$$

$$x_{ijqp} = 1 \text{ or } 0 \tag{7}$$

$$y_{iqp} = 1 \text{ or } 0 \tag{8}$$

The objective function (1) strives to minimise the total travel cost of each vehicle within a depot. The constraint (2) is to restrict that the total demand of the customer in the vehicle route should not exceed the vehicle's capacity. Equation (3) shows that the number of customers serviced by each vehicle must not exceed the number of customers a depot can serve. Equation (4) indicates that the sum of customers served by all route must be equal to the sum of customers which is served by depot *m*. Each customer serviced by a depot is shown in Equation (5). Each customer is serviced exactly once is given by Equation (6). The values of decision variable are shown in Equations (7) and (8).

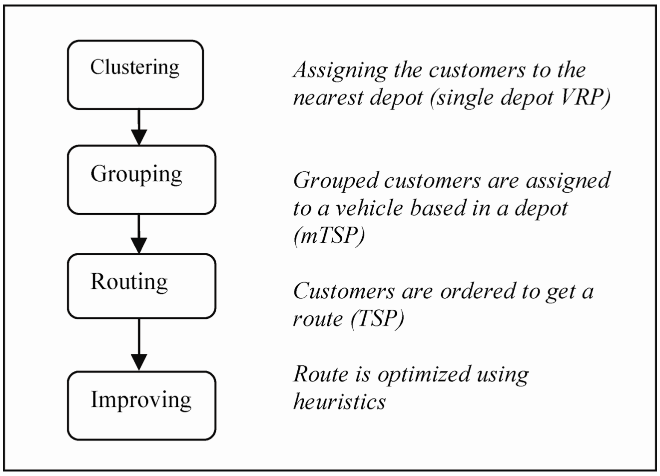
4 Proposed framework and algorithm for MDVRP

The framework used in this research consists of the following phases as shown in Figure 1.

- *Clustering phase* in which customers are generally assigned to the depot, reducing MDVRP to *m* independent VRP problem using minimax principle. The clustering also uses first-fit decreasing algorithm to utilise the capacity of vehicles to maximum and also to pack within the given number of vehicles.
- *Grouping phase* in which the customers within each depot are assigned to any one of *k* vehicles based at that depot. The assignments are made with the constraint (2).
- *Routing phase* in which minimum cost route should be identified for each vehicle similar to travelling salesman problem (TSP).

- *Improvement phase* in which local exchange 2-opt is used for improving the obtained vehicle routes.

Figure 1 Framework of proposed approach



4.1 Proposed algorithm for MDVRP

The customers are clustered into *m* groups using *k*-means algorithm. The main advantage of this algorithm is its simplicity and speed which allows it to run on large data sets. It minimises the intra-cluster variance. An additional constraint of capacity constraint is added to bring the fairness among the depot/vehicles (Geetha et al., 2009). NPSO is applied within each cluster. Master PSO assigns the customers to a vehicle based on vehicles capacity. The customers assigned to each vehicle with in the depot will not represent the routing order. So, each assigned groups is then solved as TSP using permutation encoded hybrid particle swarm optimisation (HPSO) (Geetha et al., 2010) called as slave PSO. To improve the convergence of slave PSO, the NNH is included to create initial feasible particles. MDVRP is solved as hybrid algorithm of improved *k*-means and NPSO. The output of *k*-means algorithm is pipelined as input for NPSO.

4.1.1 Improved k-means clustering algorithm

The improved *k*-means algorithm includes a priority measure to select the customer for a cluster. The customer is assigned to the nearest cluster based on maximum demand and minimum distance so that the customer having larger demand is assigned to the cluster first and the customer with smaller demand can be easily packed in to other clusters. If customers are assigned based on distance alone, the number of clusters formed may not be optimal since customers with smaller demand may be assigned to the cluster before the customer with larger demand, which may lead to the formation of additional cluster which in turn increases the number of vehicles. The concept of *minimax* principle is used for clustering the customers around the current depot. Initially *m* depots are selected as centroids. Then, the customers are arranged in decreasing order based on their demand; $q_1 > q_2 > q_3 > \ldots > q_n$. The Euclidean distance (cost_{ij}) is used in calculating the distance between the customer and the centroid as in Equation (9).

$$\text{cost}_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{9}$$

where x and y refer the coordinate in the Euclidean plane and i refers the customer and j refers the cluster. The cost_{ij} is calculated for all i to every j. Then a customer is assigned to the centroid j (closest centroid), based on the capacity constraint (2), otherwise the customer is assigned to next closest centroid.

Improved k-means clustering algorithm:

Input:

Coordinates (x_i, y_i)

Demands q_i

Customer c_i

Output:

m clusters

Procedure**:**

Select m depots as initial centroid

while not converged

for each customer $c_i \in C$,

while c_i is not assigned

Calculate the Euclidean distance measure using (9) to each of the m clusters and arrange it in sorted order

Nearest centroid of c_i is assigned as p

Assign c_i to their nearest centroid without violating the constraint (2).

if c_i is not assigned then

Choose the next nearest centroid

end if

end while

end for

Calculate the new centroid from the formed clusters using $X_j = \frac{\sum_{l=1}^{n_l} x_{c_l}}{n_j}$

and $Y_j = \frac{\sum_{l=1}^{n_l} x_{c_l}}{n_j}$ where c_l is set of customers assigned to jth cluster

end while

4.1.2 NPSO for MDVRP

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 dimension of particle is N and the number of particle is M. The ith particle is represented by $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called personal best (*pbest*). The *pbest* solution is represented as $P_i = (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{in})$. The best out of the *pbest* is called as global best (*gbest*). It is represented as $P_g = (p_{g1}, p_{g2}, p_{g3}, \ldots, p_{gn})$. The velocity rate of the particle is $V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{in})$. In the process of iteration, particles update their own velocities and positions using Equations (10) and (11).

$$v_{ij}(t+1) = wv_{ij}(t) + c_p r_1 \left[p_{in}(t) - x_{ij}(t) \right] + c_g r_2 \left[p_{gj}(t) - x_{ij}(t) \right] \tag{10}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{11}$$

$1 \leq i \leq M$, $1 \leq n \leq N$, c_p, c_g are learning factors, rand() is a random number between (0, 1) and w is a inertia factor. The pseudo-code for the PSO is as follows:

Procedure for PSO

Initialize particle

Repeat while maximum iterations or minimum error criteria is not attained

For each particle

Calculate fitness value

If the fitness value is better than the previous best fitness value

set current value as the new *pbest*

End for

Choose the particle with the best fitness value of all the particles as the *gbest*

For each particle

Calculate particle velocity according Equation (10)

Update particle position according Equation (11)

End for

4.1.2.1 Initialisation and improvement

Master PSO

The particles are coded as an integer string of length n_i (n_i is size of ith cluster). Each particle takes an integer value to represent the truck number from the set V {1 to m}. The particle position represents the corresponding customer to be serviced by the truck. Likewise, all customers are assigned to a particular vehicle to form m group within each cluster. For example, consider 5(= n_i) customers to be serviced by 2(= m) vehicles. Then, the particle in Figure 2 means the customers are grouped as {1, 2, 4} and {3, 5} for vehicles 1 and 2.

Then the groups formed are checked for the constraint (2), if violated, they are converted to the feasible solution using random swap between the groups.

Figure 2 Particle representation in master PSO

Position:	1	2	3	4	5
Particle:	1	1	2	1	2

Slave PSO (HPSO)

The feasible solutions are generated using NNH within each groups formed by master PSO. The distance matrix is calculated among all pairs of customers and between each depot and all customers using Euclidean distance metric. Initially, half the population are generated using NNH and remaining particles are randomly generated. The groups formed {1, 2, 4} and {3, 5} are taken as input for the HPSO. Then, the groups are solved for finding the sequence such that the route cost is minimum. The routes generated are optimised using 2-opt local exchange heuristics algorithm. The routes ($R1$, $R2$) are then formed by adding depot d_1 as the start and end node of each group as $R_1 - \{d_1\ 1\ 4\ 2\ d_1\}$, $R_2 - \{d_1\ 3\ 5\ d_1\}$ of D_1. Similarly, it is done for each cluster to form the route of each vehicle within the depots.

4.1.2.2 Evaluation The fitness is the summation of route cost of all vehicles in each depot. Let RC_{mk} be the route cost of kth vehicle of mth depot. Then, total cost (fitness) is calculated as:

$$\text{cost} = \sum_m \sum_k \mathrm{RC}_{mk} \tag{13}$$

If the total cost is minimum, the fitness is high. The cost of a single vehicle route of ith depot is calculated as:

$$\mathrm{RC}_{i1} = D(d_i, c_1) + \sum D(c_j, d_i) + D(c_n, d_i) \quad \text{for all } j \in n_{jq} \tag{14}$$

4.1.2.3 Genetic operations To have better exploration of huge solution space of MDVRP, the genetic operator's crossover and mutation are incorporated into PSO. The two-point crossover is applied for randomly selected particle with *gbest* and *lbest* particles. The swap mutation is used for a randomly selected particle.

4.1.2.4 Particle conversion Every solution represents the vehicle route of all depots. It should be converted to particle position value using Equation (15).

$$x_{ij} = x_{\min} + \frac{x_{\max} - x_{\min}}{n}\left(y_{ij} - 1 + r_1\right) \tag{15}$$

y_{ij} is the jth dimension of the ith solution, x_{ij} is the jth dimension of ith particle, rand() is uniformly distributed in [0, 1], n represents the number of customers, $x_{\min}$ and $x_{\max}$ are the boundary values of the particle position. For five 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 Equation (15) as shown in Figure 3.

These continuous particle position values $X_i = [x_{i1}, x_{i2}, x_{i3}, ..., x_{id}]$ are converted back to permutation of solution $Y_i = [y_{i1}, y_{i2}, y_{i3}, \ldots, y_{id}]$ using rank order value (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. Figure 4 illustrates the ROV of a particle to a permutation encoding.

Figure 3 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

4.1.2.5 NPSO algorithm for MDVRP

Input:

Coordinates (x_i, y_i)

Demands q_i

Customer c_i

m clusters

k vehicles

Output:

Vehicle routes within each cluster

Procedure:

For each cluster

Initialize the PSO Parameters

Generate the initialize solution of master PSO

While not Terminating condition met

For all solution

Decode the solution into groups

If capacity constraint (2) violated

convert it into feasible solution by swapping among the vehicle groups

End if

For each group

Call Slave PSO procedure

End for

End for

End while

End for

Procedure for slave PSO (HPSO)

Generate the initial particles for HPSO to find the routes

Evaluate the fitness using (14)

Update the *pbest,* and *lbest*

Choose the particle with the best fitness value of all the particles as the *gbest*

For all solution

Convert the real coded solution to particle position value as in Figure 3 using (15)

Calculate particle velocity according to Equation (10)

Update particle position according to Equation (11)

Apply GA operators with the probability of p_c and p_m

Convert particle back to permutation encoding as in Figure 4

Apply 2-opt local exchange

End for

Display the *gbest* as a best solution

Figure 4 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

4.1.2.6 Local exchange The common local improvement procedure 2-opt is used in the decoding method of slave PSO 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, 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.

Algorithm for 2-opt

Procedure:

For $i = 1, 2, 3, \ldots, n$ and $j = 1, 2, 3, \ldots, m$

Set n_j = number of customers in the route r_{ij}

For $p = 1, 2, \ldots, (n - 2)$ and $q = (p + 2), \ldots, n$

Modify route by changing the route direction of customer in the sequence number p, $p + 1$ and q, $q + 1$

Evaluate the routing cost

If improves modify the route, else return the route without modification

End for

End for

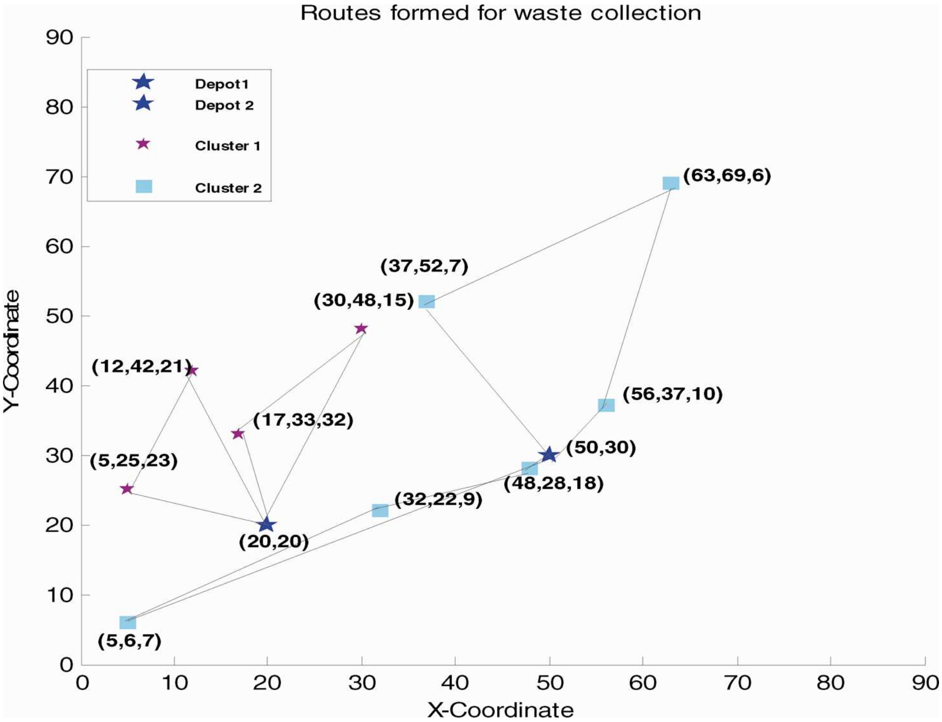
5 Case studies

Two case studies are considered for testing MDVRP applications. The garbage collection and home delivery pharmacy are the real-time applications considered for MDVRP.

The first case study deals with MDVRP which arises to address the real-life waste collection problem. It is a kind of reverse logistics in which the domestic waste is collected and disposed to a centre where it can be recycled. Usually more number of disposable centres is available for a group of collection point. The collection points are considered as customers and the disposable centres are considered as depots. The ten random collection points are generated with two disposable centres as shown in Figure 5. The coordinates of each node represent (x, y) location in Euclidean plane with third representing the amount of waste at each location. The depots are given with (x, y) location, each with two garage truck of capacity 50 units. The collection points are clustered into two to the nearest disposable centres and then the garage truck from the centres routed to the collection point to collect waste is shown in Figure 5.

The second case study is made on application of MDVRP for distribution logistics. An application of home delivery pharmacy programme is considered as a real-time example for distribution of medicines. In home delivery pharmacy, they maintain high standards of customer care, quality and accuracy. The home delivery pharmacy programme is a convenient alternative to a retail pharmacy. Customers can order through mail or phone. The medicines are delivered promptly and confidentially to the customers residence or other preferred location. The pharmacy has number of distributers located at different places. When a mail or phone is received, they allocate it to the nearby distributors to deliver the medicines to the customers. The distributors have more number of door delivery employees to serve the customers promptly. This is viewed as MDVRP, where each distribution centre/distributors are viewed as depots and the people delivering the goods are viewed as vehicle based at each depot. All employees and distributors are equally loaded with the customer demand by brining the fairness into play by clustering. Otherwise, one employee will service more customer than the other.

Figure 5 Routes for garage truck (see online version for colours)



6 Analysis of proposed algorithm

The working principle of general k-means is based on the minimum distance to the centroid. But in improved k-means algorithm, the demand is also considered for clustering the customers to have the capacity constraint as well as using the vehicles to its maximum capacity. The customers are not considered in random order instead the customers are arranged in the decreasing order of demands. This uses first-fit decreasing order and minimax principle. Because of this, a lemma is derived as follows:

Lemma 1: *Improved k-means algorithm utilises the vehicles to its maximum capacity if the customers are selected in the decreasing order of demand.*

Proof: Let C be a set of customers $\{c_1, c_2, c_3, ..., c_n\}$ and their demands $\{q_1, q_2, q_3, ..., q_n\}$. The number of clusters to be formed is m representing the depots as $\{d_1, d_2, d_3, \ldots, d_m\}$, each depot having k vehicles with uniform capacity Q. For clustering, the capacity of a depot is calculated as $k*Q$. The cost_{ij} denotes the Euclidean distance between customer c_i and depot d_j.

Suppose without loss of generality that the available customers are arranged in decreasing order of demand as, $q_1 > q_2 > q_3 > \ldots > q_n$ such that the higher demand customers are accommodated first leaving less space in the vehicle. This space is mostly enough for the customers lying towards end having minimum demands. Let (n_1, n_2, ..., n_m) be the clusters formed by improved *k*-means algorithm, where each cluster having the customers c_i if

$$\sum_{i=1}^{n_j} qc_i = k * Q, \quad \text{for } j = 1 \text{ to } m$$

$$\sum_{i=1}^{n_j} qc_i + \varepsilon = k * Q \quad \text{for } j = 1 \text{ to } m$$

where $\varepsilon \cong 0$. Otherwise, the customer c_i is considered for next nearest cluster/depot.

Suppose the customers are considered in random order, customers in the beginning may be demanding minimum are accommodated into the vehicles leaving space which may not be enough for high demanding customer coming towards end. Let (h_1, h_2, …, h_m) be the clusters formed by *k*-means without considering the sorted order of customers then,

$$\sum_{i=1}^{n_j} q_{c_i} = k * Q, \quad \text{for } j = 1 \text{ to } m$$

$$\sum_{i=1}^{n_j} q_{c_i} + \varepsilon = k * Q \quad \text{for } j = 1 \text{ to } m$$

where $\varepsilon > 0$. Hence, the vehicles are accommodated to its maximum capacity leaving less space as wastage.

6.1 Theoretical analysis

As stated, the complex NP-hard problem MDVRP is decomposed into sub-problems which are relatively solvable. Let *m* be the number of depots, *n* is the number of customers and *k* is the number of vehicles that each depot can have for servicing the customers. The clustering heuristics scans each initial solution for *m* depots to form *m* clusters. Then, for a single particle/particle the clustering takes $O(mn)$. Grouping and routing are used for assigning customers to vehicle. This is performed again for all *m* depots each with *k* vehicles and for *n* customers, which is $O(mkn)$. The improvement of routes formed which uses 2-opt local exchange takes $O(n^2)$ to perform pair-wise swapping. Thus, the exponential time complexity algorithm of MDVRP is broken into sub-problems of polynomial time complexity. The complexity of the proposed method is given in Table 1.

Table 1 Complexity of algorithm

Heuristics	*Complexity of algorithm*
Clustering	$O(mn)$
Grouping/routing	$O(mkn)$
Improvement	$O(n^2)$

6.2 Empirical analysis

The above proposed NPSO algorithm is implemented in MATLAB 7.0.1 Pentium 4.0, 2.3 GHz. The parameters of PSO used for this implementation are number of particles $I = 200$, 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 $c_g = 0.5$, local best position acceleration constant $c_l = 1.5$ with the neighbour constant $K = 5$. The mutation probability rate is $p_m = 0.4$ for performing swap or inversion mutation. The elitism is 20%. The $x_{\min} = 0.1$ and $x_{\max} = 0.7$ are values used for converting solution to particle.

The problems from Christofides and Eilon (1969), Gillet and Johnson (1976) and Chao et al. (1993) were used to validate the performance of the proposed NPSO. The main characteristics of these test problems are summarised in Table 2. These instances and the best-known solutions are available at http://neo.lcc.uma.es/radiaeb/WebVRP//index.html?/ProblemInstances /MDVRPInstances .html.

The results obtained using the proposed NPSO are compared with other heuristic technique are depicted in Table 3.

The percentage of deviation is calculated as:

$$\%\text{ of deviation} = \frac{\text{obtained solution} - \text{optimal of best-known solution}}{\text{optimal or best-known solution}} \times 100$$

Even though there is a deviation from best known solution (BKS) and compared to other heuristic the results are high, the load and route are balanced between depots and vehicles of a depot. Tables 4 and 5 show the comparative results for the load and route balance of depots with respect to BKS.

The results show that the load between the depots is well balanced as the deviation is smaller when compared to the BKS. The route balancing is calculated as:

$$\text{Route balance} = \text{maximum route length} - \text{minimum route length}$$

Table 2 Characteristics of test problems

Problem	*No. of depots*	*No. of customers*	*Capacity of vehicle*
1	4	50	80
2	4	50	160
3	5	75	140
4	2	100	100
5	2	100	200
6	3	100	100
7	4	100	100
12	2	80	60
13	2	80	60
14	2	80	60
15	4	160	60
16	4	160	60
17	4	160	60

The route is also balanced between the depots using NPSO when compared to BKS. The results presented in Tables 4 and 5 show that the problems for which load are not properly balanced are reflected for route balancing also. As stated in Section 5, the load is balanced between the depots by means of *k*-means algorithm. The NPSO balances the route and load of each vehicle of a depot.

Table 3 The results of proposed method compared with other heuristic methods

Problem instances	*BKS*	*FIND (Renaud et al., 1996)*	*CGL (Cordeau et al., 1997)*	*GenClust (Thangiah and Salhi, 2001)*	*NPSO*	*% Deviation from BKS*
1	576.86	576.86	576.86	591.73	610.74	5.87
2	473.53	473.53	473.87	463.15	507.67	7.21
3	641.18	641.18	645.15	694.49	679.1	5.91
4	1,001.49	1,003.86	1,006.66	1,062.38	1,102.6	10.09
5	750.26	750.26	753.4	754.84	821.43	9.49
6	876.5	876.5	877.84	976.02	977.5	11.52
7	885.69	892.58	891.95	976.48	987.25	11.47
12	1,318.95	1,318.95	1,318.95	1,421.94	1,602.1	21.47
13	1,318.95	1,318.95	1,318.95	1,318.95	1,342.6	1.79
14	1,360.12	1,365.68	1,360.12	1,360.12	1,387.4	2.0
15	2,505.29	2,551.45	2,534.13	3,059.15	3,106.2	23.99
16	2,572.23	2,572.23	2,572.23	2,719.98	3,005.5	16.84
17	2,708.99	2,731.37	2,720.23	2,894.69	3,085.5	13.9

Table 4 Standard deviation (SD) of load balance for BKS and NPSO

S. no.	*SD for BKS*	*SD for NPSO*
1	74.9	44.21
2	77.84	44.21
3	74.07	47.81
4	56.5	69.29
5	83.4	69.29
6	107.5	89.01
7	30.1	81.26
12	0	0
13	0	0
14	0	0
15	0.81	0
16	8.08	0
17	0	0

Table 5 Route balance for BKS and NPSO

S. no.	*Route balance of BKS*	*Route balance of NPSO*
1	162.10	89.82
2	82.5	73.29
3	139.80	59.6
4	63.1	95.9
5	6.4	0.70
6	220.6	217.4
7	45.8	75.99
12	0	8.5
13	0	8.5
14	13.3	7.5
15	41.13	20.5
16	0	15.21
17	13.25	15.35

7 Conclusion

In logistic distribution, routing and scheduling are the two main operational decisions. The total cost of delivery is lowered only by better routing and scheduling. As already stated, in many real-time applications VRP fails to work. Generally, there exist many depots each with limited number of vehicles for servicing. In this research work, a new framework for PSO is proposed and called as NPSO.

In this paper, MDVRP is solved in four levels: clustering the customers to the depots, assigning the vehicle for customer, finding the route for a vehicle and improving the route obtained. If MDVRP is implemented as different independent parts, the output of one cannot be given as input to clustering phase. But incorporating all levels into a single phase, the information can flow from one level to other level in search for better result. Because the MDVRP is integration of two or more NP-hard optimisation problems, a hybrid algorithm is proposed in this work. The algorithm combines clustering and PSO as nested algorithm. This combination of heuristics works better for solving MDVRP. MDVRP is relatively a complex problem when compared with VRP. It is partitioned into sub-problems of polynomial time complexity. Although the results when compared with BKS are not promising, this work focuses in resolving the cost incurred between the depots/vehicles load. The objective is focused in distributing the load and route between the depots/vehicles uniformly. It has been applied to the real-world problems such as home delivery of goods and waste collection problem and discussed as case study. From the results obtained, it is clear that pre-processing of data is best rather than random generation of population. Further, the new NPSO algorithm can also be used for solving multi-depot location routing problem, *p*-median problem, etc. and for many real-time applications. The NPSO can be further extended to MDVRP with time window by adding the constraints.

Acknowledgement

We thank the anonymous reviewers for their valuable comments, which help to enhance our paper in a better way.

References

Ai, T.J. and Kachitvichyanukul, V. (2007) 'Particle swarm optimization for the capacitated vehicle routing problem', *Int. J. Logistics and SCM Systems*, Vol. 2, pp.50–55.

Ai, T.J. and Kachitvichyanukul, V. (2009a) 'A particle swarm optimization for vehicle routing problem with time windows', *Int. J. Operational Research*, Vol. 6, Nos. 3–4, pp.519–537.

Ai, T.J. and Kachitvichyanukul, V. (2009b) 'Particle swarm optimization and two solution representation for solving the capacitated vehicle routing problem', *Computer and Industrial Engineering*, Vol. 56, pp.380–387.

Ball, M., Golden, B., Assad, A. and Bodin, L. (1983) 'Planing for truck fleet size in the presence of a comman carrier option', *Desicion Science*, Vol. 14, pp.103–120.

Chao, I.M., Golden, B.L. and Wasil, E. (1993) 'A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions', *American Journal of Mathematical and Management Science*, Vol. 13, pp.371–406.

Christofides, N. and Eilon, S. (1969) 'An algorithm for the vehicle routing dispatching problem', *Operational Research*, Vol. 20, No. 3, pp.309–318.

Cordeau, J.F., Gendreau, M. and Laporte, G. (1997) 'A tabu search heuristic for periodic and multi-depot vehicle routing problems', *Networks*, Vol. 30, pp.105–119.

Dantzig, G.B. and Ramset, J.H. (1958) 'The truck dispatching problem', *Management Science*, Vol. 6, pp.81–91.

Eberhart, R.C. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. Piscataway, NJ and Nagoya, Japan: IEEE Service Center, pp.39–43.

Geetha, S., Poonthalir, G. and Vanathi, P.T. (2009) 'Improved K-means algorithm for capacitated clustering problem, INFOCOMP', *Journal of Computer Science*, Vol. 8, No. 4, pp.52–59.

Geetha, S., Poonthalir, G. and Vanathi, P.T. (2010) 'A hybrid particle swarm optimization with genetic operator for vehicle routing problem', *Journal of Advnces in Information Technol*ogy, Vol. 1, No. 4, pp.181–188.

Gillet, B. and Johnson, J. (1976) 'The multiple terminal vehicle dispatching algorithm', *Omega*, Vol. 4, pp.711–718.

Giosa, I.D., Tansini, I.L. and Viera, I.O. (2002) 'New assignment algorithms for the multi-depot vehicle routing problem', *Journal of the Operational Research Society*, Vol. 53, pp.977–984.

Kennedy, J. and Eberhart, R.C. (1995) 'Particle swarm optimization', *Proceedings of the IEEE International Conference on Neural Networks*, Vol. IV. Piscataway, NJ: IEEE Service Center, pp.1942–1948.

Klots, B., Gal, S. and Harpaz, A. (1992) 'Multi-depot and multi-product delivery optimization problem with time and service constraints', *Israel Technical Representation*, IBM Israel, Haifa.

Lim, A. and Wang, F. (2005) 'Multi-depot vehicle routing problem: a one stage approach', *IEEE Transaction on Auotimation Science and Engineering*, Vol. 2, No. 4, pp.397–402.

Mao-Xiang, L. (2006) 'Study on the model and algorithm for multi-depot vehicle schedling problem', *Journal of Transportation Systems Engineering and Information Technology*, Vol. 16, No. 15, pp.65–70.

Mingozzi, A. (2005) 'The multi-depot periodic vehicle routing problem', *Proceedings of SARA'2005*, pp.347–350.

Nilay, Y.G. and Nihan, Ç.D. (2011) 'A new geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem', *Expert Systems with Applications*, Vol. 38, No. 9, pp.11859–11865.

Pongchairerks, P. and Kachitvichyanukul, V. (2009) 'A two-level particle swarm optimization algorithm on job-shop scheduling problems', *Int. J. Operational Research*, Vol. 4, No. 4, pp.390–411.

Potvin, J. and Rousseau, J. (1993) 'A parallel route building algorithm for the VRPTW', *European Journal of Operational Research*, Vol. 66, pp.331–340.

Raft, O.M. (1982) 'A modular algorithm for an extended vehicle scheduling problem', *European Journal of Operational Research*, Vol. 11, pp.67–76.

Renaud, J., Laporte, G. and Boctor, F.F. (1996) 'A tabu search heuristic for the multi-depot vehicle routing problem', *Computers & Operations Research*, Vol. 23, pp.229–235.

Sombuntham, P. and Kachitvichayanukul, V. (2010) 'A particle swarm optimization algorithm for multi-depot vehicle routing problem with pickup and delivery requests', *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010*, Vol. 3, Hong Kong, 17–19 March, 2010.

Su, C.T. (1999) 'Dynamic vehicle control and scheduling of a multi-depot physical distribution system', *Integrated Manufacturing Systems*, Vol. 10, pp.56–65.

Thangiah, S.R. and Salhi, S. (2001) 'Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem', *Applied Artificial Intelligence*, Vol. 15, pp.361–382.

Tillman, F.A. (1969) 'The multiple terminal delivery problem with probabilistic demands', *Transportation Science*, Vol. 3, pp.192–204.

Wenjing, Z. and Ye, J. (2010) 'An improved particle swarm optimization for the multi-depot vehicle routing problem', *ICEE, 2010 International Conference on E-Business and E-Government*, pp.3188–3192.

Yu, B., Yang, Z-Z. and Xie, J-X. (2011) 'A parallel improved ant colony optimization for multi-depot vehicle routing problem', *Journal of the Operational Research Society*, Vol. 62, No. 1, pp.183–188.

Zhang, K., Tang, J. and Fung, R.Y.K. (2011) 'A scatter search for multi-depot vehicle routing problem with weight-related cost', *Asia-Pacific Journal of Operational Research*, Vol. 28, No. 3, pp.323–348.

Zhen, T. and Zhang, Q.W. (2009) 'A combining heuristic algorithms for the multi-depot vehicle routing problem with inter-depot routes', *International Joint Conference on Artificial Intelligence*, pp.436–439.