A Particle Swarm Optimization Algorithm for Multi-depot Vehicle Routing problem with Pickup and Delivery Requests

Pandhapon Sombuntham and Voratas Kachitvichayanukul

Abstract— **A particle swarm optimization algorithm with multiple social learning structures is proposed for solving the practical case of multi-depot vehicle routing problem with simultaneous pickup and delivery and time window. In the problem of interest, each location may have goods for both pickup and delivery with multiple delivery locations that may not be the depots. An extension of GLNPSO with the new decoding procedure is proposed. Computational experiments are carried out using the test instances for the pickup and delivery problem with time windows (PDPTW) as well as a newly generated instance. The preliminary results show that the proposed algorithm is able to provide the good solutions to some of the problems.**

Index Terms— **vehicle routing problem, pickup and delivery, multiple depot, particle swarm optimization**

I. Introduction

The distribution of products from depots to customers is one of the key activities that play an important role in the effectiveness of business. In general, the problem is known as the vehicle routing problem (VRP). In VRP, there is a set of customers to be served by a set of vehicles from a depot. The objective is to determine the optimal sequence of customers visited by each vehicle, called vehicle route, which satisfies the certain criteria such as distance, time, and cost involved in the operation.

Many variants of VRP are studied to address the variety of conditions in real world applications. For example: the capacitated VRP (CVRP), the VRP with time windows (VRPTW), the heterogeneous fleet VRP (HVRP), the VRP with pickup and delivery (VRPPD), and multiple depots VRP (MDVRP).

VRPPD is the generalized version of VRP which involves not only the deliveries but also the pickups of commodities from customers[1].The VRPPD can be further subdivided into delivery-first and pickup-seconds, mixed pickups and deliveries, and simultaneous pickups and deliveries, (see [1].) Reference [2] has classified the pickup and delivery problems (PDP) into three different groups based on the pickup-delivery relation. The first one is many-to-many problem. This is the case when any node can serve as a source or destination for any commodity. A commodity may be picked up from one of many locations, and also delivered to one of many locations. The second one is one-to-many-to-one problem. In this case, commodities are initially available at a depot and must be delivered to the customers, and, in addition, the commodities available at the customers must be delivered to the depot. This characteristic is usually found in many studies regarding VRPPD. The last one is one-to-one problem. Each commodity, can be called a request, has a given origin and a given destination. The variants of VRP and PDPs involve many constraints and are known to be NP-hard problems which consume a lot of computational time. The heuristic approach such as tabu search algorithm, ant colony optimization, genetic algorithm (GA), and particle swarm optimization algorithm (PSO) are commonly adopted to solve these problems. These methodologies do not guarantee optimal solutions, but they could promise a near optimal solution in a reasonable time.

An approach called reactive tabu search was developed by Nanry and Barnes [3] to solve PDP with time window (PDPTW). Some classical Traveling Salesman Problem (TSP) heuristics were adapted to deal with PDP which has a single vehicle [4]. Three sets of heuristics algorithm were developed to deal with the VRPDD which cover the case of mixed and simultaneous pickup and delivery VRP [1], which can be viewed as one-to-many-to-one PDP. Ropke and Pisinger [5] proposed an adaptive large neighborhood search (ALNS) heuristics to handle PDPTW. The aim of their study is to minimize total distance; times spent by each vehicle, and maximize fulfilled demand. At each iteration, three removals and two insertions are used to rearrange some of the requests. Ai and Kachitvichyanukul ([6], [7], [8]) have applied a real-value version of PSO for solving CVRP, VRP with simultaneous pickup and delivery, and VRPTW. The PSO algorithm used a solution representation which is consisted of the priority list of customer and its preferred vehicle. The particle is converted into the problem specific solution through decoding procedure. The studies present two solution representations, SR-1 and SR-2, for both cases.

A. Problem description

Though, in many study, a *customer* is usually refers to as a place that must be served by a vehicle from a depot. Here, a more general term *location* is used instead of customer as a location can also supply goods to others by using vehicle from other sources. The literatures regarding VRPPD and PDP are usually focused on the case for which each location,

Manuscript received December 30, 2009. This work was supported in part by Thailand Research Fund under Grant MLSC525009.
P. Sombuntham is a master student in Industrial and Manufacturing Engineering, Asian institute of Technology, Pathumthani, 12120 Thailand (phone:(662)524-6135; fax:(662)524-5697; e-mail: pandhapon.sombuntham@ ait.ac.th).
V. Kachitvichyanukul is a professor of Industrial and Manufacturing Engineering, Asian institute of Technology, Pathumthani, 12120 Thailand. (e-mail: voratas@ait.ac.th).

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

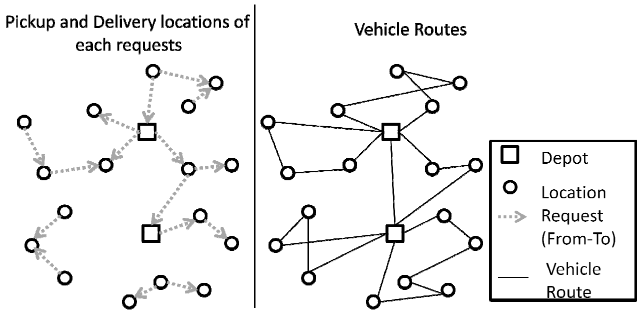

Fig.1 The VRP with many-to-many requests.

excluding depot, has only one destination for the commodities picked up at its location. It also can be found that many studies in VRP with pickup and delivery (VRPDD) had made rigid assumptions on the destination of the pickup commodities. The product generally destined to be brought back to the depot. For PDP, a location is usually either a pickup or delivery location of a request. However, in real-world situation, a certain location can be either pickup location, delivery location or both. Moreover, goods pickup from that location can be destined to several other locations as shown in Fig.1. The application of this case can be found in alliance of Small and Medium-sized Enterprises (SME) where the firms may not have sufficient fund to invest in transportation resources and it is essential to form a business alliance to pool their resources. The common resources they may share are the fleet of vehicles which are used to transport goods among their alliance members. As a result, some locations may act like depots without owning fleet of vehicles.

This study presents an approach to handle the practical case of pickup and delivery of goods as stated earlier. Each location can have many associated requests destined to be delivered to various locations. Time window, heterogeneous fleet of vehicles, and multiple depot characteristics are also taken into account in this study. Hence, the problem is denoted here as the generalized vehicle routing problem for multi-depot with pickup and delivery requests (GVRP-MDPDR). The objectives are to minimize both total distance and number of vehicle used while simultaneously maximize the number of fulfilled requests. The proposed method is based on the particle swarm optimization algorithm GLNPSO which is a version of PSO algorithm with multiple social learning structures [10, 11].

B. *Particle swarm optimization*

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart [9]. In PSO, a solution of a specific problem is being represented (directly or indirectly) by an n-dimensional position of a particle. The search is performed by moving the particle to a new position via a velocity vector. The PSO algorithm starts with a population of particles initialized with random position and velocity. The population of particles is usually called a swarm. In one iteration step, every particle is moved from previous position to the new position based on its velocity. The velocity of a particle is updated based on the particle's personal best position, called *pbest*, and the global best position found so far by the swarm, namely *gbest*. This allows particles to exchange their experience to ensure the diversity of the search and lead to improvement of solutions.

A PSO framework for solving GVRP-MDPDR is based on GLNPSO, a PSO algorithm with multiple social learning structures, see [10] and [11]. The real-value PSO is used to construct customer priority list and vehicle priority matrix through encoding and decoding method. Encoding method defines how a particle is constructed and it is usually called the solution representation. The method used to convert a particle to a problem specific solution is called decoding method. The decoding method used in this study is based on the SR-1 of [7] with some neighbor moves added to construct the solution.

The remainder of this paper is organized as follow: Section 2 presents the problem formulation in term of mathematical model. Section 3 presents the PSO framework which is based on [7], [10], and [11]. Section 4 explains the computational example of the proposed methodology. Finally, Section 5 summarizes the result of this study together with suggestion for further study.

II. Problem Formulation

The objectives considered in this study are total distance, number of vehicles used, and number of fulfilled request. The vehicle routes are formed in such a way that

(1) the total routing cost is minimized;
(2) the number of vehicle is minimized;
(3) the fulfilled demand is maximized.

In addition, the following restrictions must be met:

(4) each request is served exactly once by a vehicle;
(5) the load of a vehicle never exceeds its capacity;
(6) each route starts and ends at the indicated terminals;
(7) the number of vehicles used do not exceed maximum number of available vehicles;
(8) the total duration of each route (including travel and service time) does not exceed a preset limit.

The mathematical model for the problem is based on the model found in [5]. The main differences lie in the objective function and all vehicles are allowed to serve any requests if the assigned request does not exceed capacity. Moreover, same location can have multiple requests, pickup nodes and delivery nodes can share same x-y coordination. The model served as a formal description of the problem and is given below.

A. *Input Parameter*

P = $\{1,\dots, n\}$ is the set of pickup nodes,

D = $\{n+1,\dots, 2n\}$ is the set of delivery nodes.

N $=P \cup D$ is set of all pickup and delivery nodes

H_i is a penalty cost if the request i is not served, $i \in P$.

K is the set of all vehicles, $|K| = m$.

C_k is the capacity of vehicle $k \in K$.

f_k is the fixed cost of vehicle $k \in K$ if it is used.

τ_k is the nodes that represents the start terminal of vehicle k, $k \in K$

τ'_k is the nodes that represents end terminal of vehicle k, $k \in K$

$V = N \cup \{\tau_1,\dots,\tau_m\} \cup \{\{\tau'_1,\dots,\tau'_m\}$ is set of all nodes.

A is a set of (i, j) which is an arc from node i to node j, where $i, j \in V$.

d_{ij} and t_{ij} are nonnegative distance and travel time between node i and node j, for i and j $\in N$. Travel

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

times satisfy the triangle inequality; $t_{ij} \leq t_{il} + t_{lj}$ for all $i, j, l \in V$

s_i is a service time required for loading and unloading when visiting node i

$[a_i,b_i]$ is a time windows when the visit at the particular location must start; a visit to node i can only take place between time a_i and b_i

l_i is a value of the amount of goods that must be load onto the vehicle at node i for $i \in P$ and $l_i = -l_{i-n}$ for $i \in D$.

Request i is represented by nodes i and i+n ,where i $\in$P and i+n $\in$D, and any nodes can have same x-y coordinate as the same location can have multiple requests in this study.

B. Decision variables

The main decision variables in the model are described below:

x_{ijk} is a binary variable which is one if the edge between node i and node j is used by vehicle k and zero otherwise, where $i, j \in V$, $k \in K$.

S_{ik} is a nonnegative integer that indicates when vehicle k starts the service at location i, $i \in V$, $k \in K$

L_{ik} is a nonnegative integer that is an upper bound on the amount of goods on vehicle k after servicing node *I, where* , $i \in V$, $k \in K$. S_{ik} and L_{ik} are only well-defined when vehicle k actually visits node i.

z_i is a binary variable that indicates if request i is placed in the request bank, where , $i \in P$. The variable is one if the request is placed in the request bank and zero otherwise.

C. A mathematical model

The mathematical formulation is given below:

Minimize
$$\alpha \sum_{k\in K} \sum_{(i,j)\in A} d_{ij} x_{ijk} + \beta \sum_{k\in K} \sum_{j\in P} f_k \, x_{\tau_k,j,k} + \gamma \sum_{i\in P} H_i \, z_i \quad (1)$$

Subject to;

$$\sum_{k\in K} \sum_{j\in N_k} x_{ijk} + z_i = 1 \ , \qquad \forall i \in P \quad (2)$$

$$\sum_{j\in V} x_{ijk} - \sum_{j\in V} x_{j,n+i,k} = 0 \qquad \forall k \in K, \forall i \in P \quad (3)$$

$$\sum_{j\in P\cup\{\tau_k'\}} x_{\tau_k,j,k} = 1 \qquad \forall k \in K \quad (4)$$

$$\sum_{i\in D\cup\{\tau_k\}} x_{i,\tau_k',k} = 1 \qquad \forall k \in K \quad (5)$$

$$\sum_{i\in V} x_{ijk} - \sum_{i\in V} x_{j,i,k} = 0 \qquad \forall k \in K, \forall j \in N \quad (6)$$

$$x_{ijk} = 1 \Rightarrow S_{ik} + s_i + t_{ij} \leq S_{jk} \qquad \forall k \in K, \forall (i,j) \in A \quad (7)$$

$$a_i \leq S_{ik} \leq b_i \qquad \forall k \in K, \forall i \in V \quad (8)$$

$$S_{ik} \leq S_{n+i,k} \qquad \forall k \in K, \forall i \in P \quad (9)$$

$$x_{ijk} = 1 \Rightarrow L_{ik} + l_i \leq L_{jk} \qquad \forall k \in K, \forall (i,j) \in A \quad (10)$$

$$L_{ik} \leq C_k \qquad \forall k \in K, \forall i \in V \quad (11)$$

$$L_{\tau_k k} = L_{\tau_k' k} = 0 \qquad \forall k \in K \quad (12)$$

$$x_{ijk} \in \{0,1\} \qquad \forall k \in K, \forall (i,j) \in A \quad (13)$$

$$z_i \in \{0,1\} \qquad \forall i \in P \quad (14)$$

$$S_{ik} \geq 0 \qquad \forall k \in K, \forall i \in V \quad (15)$$

$$L_{ik} \geq 0 \qquad \forall k \in K, \forall i \in V \quad (16)$$

The objective function minimizes the weighted sum of the distance traveled (α), the sum of the fix cost for each used vehicle (β), and the penalty cost associated with number of requests not scheduled (γ).

Equation (2) ensures that each pickup location is visited or that the corresponding request is placed in the request bank. Equation (3) ensures that the delivery location is visited if the pickup location is visited and that the visit is performed by the same vehicle. Equations (4) and (5) ensure that a vehicle leaves every start terminal and a vehicle enters every end terminal. Together with equation (6) this ensures that consecutive paths between τ_k and τ'_k are formed for each vehicle $k \in K$. Equations (7) and (8) ensure that S_{ik} is set correctly along the paths and that the time windows are obeyed. These constraints also make sub tours impossible. Equation (9) ensures that each pickup occurs before the corresponding delivery. Equations (10), (11) and (12) ensure that the load variable is set correctly along the paths and that the capacity constraints of the vehicles are enforced.

Though, this study will focus on the case when one location has many associated items that must be shipped to many different locations, it is clear that the pickup and delivery node of each request in the mathematical model is modeled separately and not in integer-linear form. The first reason is that it is much easier to understand the problem description from the formulation. The second reason is that since the problem will be solved heuristically, the integer-linear form of model may not be necessary. Moreover, the model can support many variants of VRP such as VRPTW, CVRP, HVRP, VRPPD, and PDPTW.

III. PSO FOR THE PROBLEM

The main algorithm used in this paper is GLNPSO which is the variant of PSO algorithm with multiple social learning terms as proposed in [10] and [11] .The VRP specific part of the procedure is based on [6], [7], and [8]. The basic framework is the same as shown in Fig. 2. The main modifications are in the route construction procedure to meet the practical constraints and to handle the differences in the problem requirements.

For the problem with n locations and m vehicles, a swarm of particles with (n+2m)-dimension is constructed. The values of positions, velocity, and personal best of each particle are first set in the initialization step. In each iteration, the position of each particle is decoded into the vehicle route and its fitness evaluated. The personal best and social learning terms will be updated prior to the updates of the velocity and position of each particle. The social learning terms used in the GLNPSO are global best (*gbest*), local best (*lbest*), and near neighbor best (*nbest*). The algorithm repeats until the stopping criterion which generally is a number of iterations is met.

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

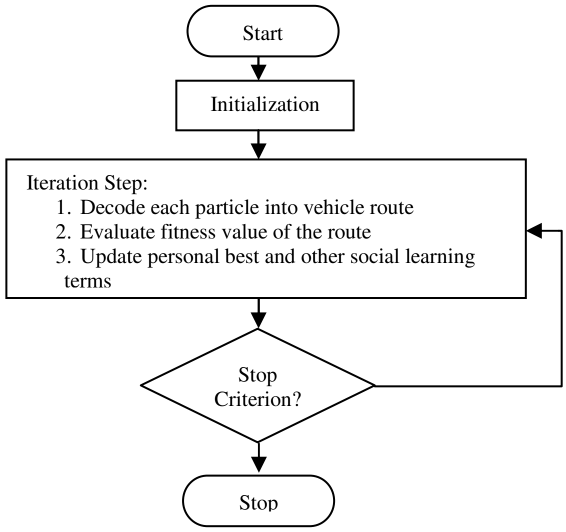

Fig.2 PSO framework for solving GVRP-MDPDR, based on [8]

A. Solution Representation

The particle representation described here is based on [6], [7], and [8]. The particle composed of two parts. For n locations and m vehicles problem, the particle will be a vector of n+2m elements. The first n dimensions of a particle represent the priority for location. The next 2m dimensions represent the x-y coordinate of the orientation points of m vehicles. The point serves as the representative of the area which the vehicle prefers to perform its duty. The vehicle with its orientation point closer to a location will be preferable when considering that location.

B. Decoding Method

Decoding method consists of three steps to convert the particle into the solution of the problem. The first part is constructing a location priority list, and the second part is constructing vehicle priority matrix. The first two parts will give a priority list of locations and preferred list of vehicles of each location. The third part is route construction by assigning requests using the location priority and vehicle priority. A request which has not been served by a vehicle is denotes as an unfulfilled request. For a problem with n customer, m vehicle, and r requests

Denote:

$N = \{N_1, N_2, \ldots, N_n\}$-set of locations, N_i is the location index with ith priority.

$L= \{L_1, L_2, \ldots, L_r\}$-set of all requests.

$W = \{W_{11}, W_{12},\ldots, W_{nm}\}$-vehicle priority Matrix ,where W_{ij} is vehicle index with j^{th} priority corresponding to location priority i^{th} .

U_p = set of unfulfilled requests which has p as a pickup location, $p = N_i$ where $N_i \in N$.

D_l = the delivery location of request l where $l \in L$.

R_c = route of vehicle c, c= 1, 2,.., m.

S_c = Set of requests assigned to vehicle c, c= 1, 2, .., m.

Decoding Algorithm

1. Constructing Location Priority List
   a. Consider the position value of the first n dimension of a particle as the corresponding position value of locations
   b. Sort the location index based on its corresponding position value. The smaller position value, the higher priority of that location.
   c. The sorted list of customer index is considered as the location priority list. (Construct N)
2. Constructing Vehicle Priority Matrix
   a. Consider the position value of next 2m dimension, n+1 to 2m, as the vehicle orientation points.
   b. For each location in the location priority list
      i. Compute Euclidean distance between the location and vehicle orientation points
      ii. Sort the vehicle index based on Euclidean distance in ascending order.
      iii. The sorted vehicle index is considered as the corresponding row for the location in the vehicle priority matrix. (Construct W)
3. Route Construction

Start with i =0and j = 0, where i = 0,1,2,…, n and j = 0,1,2,…, m

   a. Set $p = N_i$,$c = W_{ij}$, $R_c' = R_c$, $S_c' = S_c$, and $U_p' = U_p$
   b. Set k = last position of the R_c' before the terminal depot.
   c. Make a new candidate route by inserting p in the k^{th} position of the R_c'.
      i. Assign unfulfilled request to S_c' based on R_c' if feasible. Update U_p'.
      ii. Consider request l in U_p'
         1. Insert the D_l into the R_c' which yield the lowest increase in distance (Cheapest insertion) and is feasible.
         2. Remove request l from U_p' and assign request l to S_c'.
         3. Assign unfulfilled request based on R_c' if feasible. Update U_p'.
      iii. Repeat 3.c.ii until all requests in U_p' are considered
   d. Repeat 3.c until all positions are considered.
   e. For $R_{c'}$ with highest number of assigned requests, Set $R_c= R_{c'}$ and $U_p = U_p'$.
   f. If there is no more requests in U_p, move to next N_i and repeat 3.a. Otherwise, consider next W_{ij} and repeat 3.a.

In the route construction step, it can be seen that there are neighborhood moves applied to improve the route. Step 3.c is inserting a location into the route and considers the request that must be picked up from that location. Then the delivery locations of the requests are inserted into the route. It is noted that the insertion position of pickup location may also move to different position for better number of served requests and distance. Another concept applied here is that the vehicle should be loaded with as much goods as possible when it visits a certain location. In steps 3.c.i and 3.c.ii.3, the unfulfilled requests will be assigned to the vehicle if they have both pickup and delivery locations contained in the existing vehicle route. It is denoted that the insertion must not violate the capacity constraints. Following this decoding method, there is a possibility that same location will be inserted in to the same route more than once. This means that vehicles are allowed to go back for re-stocking or unloading before continue their delivery.

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C. Reduce the number of vehicles

In addition to the decoding method, an improvement procedure is also applied to reduce the number of vehicles used. The procedure is applied to all particles during the initialize steps and to the global best at some iteration. For initialize steps, every particle is decoded and evaluated. Here, number of vehicle used is reduced by one each time. The particle is then decoded again. If the removal leads to better results, the number of vehicle used is reset for all particles. The reduction continues until the particle is decoded into an infeasible solution before moving to the next particle. The same procedure of reduction also applied to the global best at certain iteration. If the number of vehicle used of the global best is reduced, those of all other particles are also reset. The result of this method is the reduction of vehicles used and the appropriate dimension of particles when number of vehicle is reduced.

IV. Computational Results

This section describes computational results to assess the performance of the proposed algorithm. First, the proposed approach is applied to solve some of 100-locations instances constructed by Li and Lim [12]. The instances are single depot pickup and delivery problem with time windows with the primary objective to minimize number of homogenous vehicle used and the secondary objective to minimize the total distance. Additional instances are generated to represent the cases of interest based on the set C101 found in [13].

A. PDPTW Instances

There are 3 types of instances being solved here, clustered locations with short schedule horizon (lc101 to lc109), clustered locations with long schedule horizon (lc201 to lc208), and randomly distributed locations with short schedule horizon (lr101 to lr112). In each case, the problem is solved 5 times to compute average number of vehicle used and average total distance obtained. Since the primary objective of the instances is to minimize number of vehicle used, the fix cost of a vehicle is set to 10,000. Set $\alpha = \beta = \gamma$ = 1 and a penalty cost, H_i, if a request is not served, to have a high value.

The PSO parameters are set as followed: 100-particles and 1000-iterations. The constant values, inertia weight, and number of neighbors are set based on the work of Ai and Kachitvichyanukul [7], i.e., 5 neighbors, inertia weight linearly decreasing from 0.9 to 0.4. The acceleration constants for personal best, global best, local best, and near neighbor best are 0.5, 0.5, 1.5, and 1.5. Each instance is run 5 times. The results are shown in Table I.

The proposed approach shows the promising result in the clustered problem with short and long schedule horizon. This may be the effectiveness from the idea of vehicle orientation point which constructs the route from close-distance location first. However, from the result of the case of randomly-distributed-geographical instances, there is still a gap for improvement. From observations, it takes more computation time for the second and third types of instances than the first type of instances. For long horizon instances, longer computing time may be due to the many available combinations. It can be seen that the randomly-distributed locations cases with long horizon schedule and half-clustered-half-random instances are not presented here as further investigations are necessary. It seems that the concept of orientation point of vehicle might not be appropriate when the locations of customers are randomly distributed.

Table I Computational result for Li & Lim‘s PDPTW

Case	Best known solution		Best of 5 Replications		Average	
	NV	Distance	NV	Distance	NV	Distance
lc101	10	828.94	**10**	**828.94**	10.00	828.94
lc102	10	828.94	**10**	**828.94**	10.00	828.94
lc103	9	1035.35	9	1063.63	9.40	977.47
lc104	9	860.01	9	863.36	9.00	884.14
lc105	10	828.94	**10**	**828.94**	10.00	828.94
lc106	10	828.94	**10**	**828.94**	10.20	890.53
lc107	10	828.94	**10**	**828.94**	10.00	828.94
lc108	10	826.44	**10**	**826.44**	10.20	839.41
lc109	9	1000.60	10	827.82	10.00	828.04
lc201	3	591.56	**3**	**591.56**	3.00	591.56
lc202	3	591.56	**3**	**591.56**	3.00	591.56
lc203	3	585.56	3	591.17	3.00	591.17
lc204	3	590.60	**3**	**590.60**	3.00	616.26
lc205	3	588.88	**3**	**588.88**	3.00	590.41
lc206	3	588.49	**3**	**588.49**	3.00	588.49
lc207	3	588.29	**3**	**588.29**	3.00	588.29
lc208	3	588.32	**3**	**588.32**	3.00	588.32
lr101	19	1650.80	**19**	**1650.80**	19.00	1661.66
lr102	17	1487.57	17	1512.25	17.00	1559.75
lr103	13	1292.68	13	1300.77	13.00	1360.04
lr104	9	1013.39	10	1050.90	10.40	1107.89
lr105	14	1377.11	14	1389.43	14.00	1397.72
lr106	12	1252.62	12	1270.46	12.40	1293.20
lr107	10	1111.31	10	1147.12	11.20	1221.63
lr108	9	968.97	**9**	**968.97**	9.20	981.59
lr109	11	1208.96	12	1287.91	12.80	1346.97
lr110	10	1159.35	11	1212.82	11.60	1242.83
lr111	10	1108.90	11	1158.45	11.00	1192.48
lr112	9	1003.77	11	1143.76	11.00	1185.72

Note: **NV** = Number of vehicles used, Bold value in column 4 and 5 indicates that it has same value as the best known solution.

Table II Computational result of MC1_1

Value	Replication				
	1	2	3	4	5
NV	32	31	31	31	31
Total distance	6597	6526.4	6707.5	7125.6	6421.6
Objective Function	9317	9196.4	9342.5	9730.6	9056.6

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

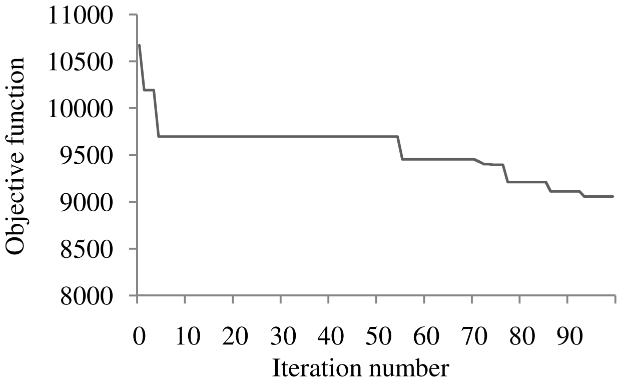

Fig. 3 Objective function and iteration number of Replication 5 of MC1_1

B. New Instances

A new instance of GVRP-MDPDR, denoted as MC1_1, is generated based on the geographical data in C101 of Solomon's data set. The original instance is a vehicle routing problem with time windows with clustered location. The instance has 100 locations without vehicles and 4 locations with 40 vehicles in total. There are 200 requests. Set $\alpha = \beta = \gamma = 1$ and a penalty cost, H_i, if a request is not served, to have a high value. The description of instances is shown as the followings.

No. of Location	: 100
No. of depot	: 4
No. of Requests	: 200
No. of Vehicle	: 40
Vehicle capacity	: 200
Fix cost of vehicle	: 75, 90 or 100
Range of Quantity of order	: 10 to 50
Min time windows range	: 85
Avg. time windows range	: 703.67
Max time windows range	: 1123

The PSO parameters are same as the previous section except number of particles and iterations which are 50 and 100, respectively. The result is shown in Table II.

It is noted that there is no unfulfilled requests in any replications as a penalty cost, H_i, if a request is not served, has high value. With the same iteration and parameters, the solution obtained from each replication is consistent. From Fig. 3, the objective function continually decrease when the iteration number increase. As the PSO can give diversity in solution and consistently maintain or improve the best solution, this may lead to near optimal solution in later iterations.

V. Conclusion

In this study, the practical case of generalized vehicle routing problem is studied as the number of destinations of pickup goods is not limited to one. Due to complexity of the problem, the real-value PSO with newly proposed decoding method with some neighbor moves is used to solve the problem. The proposed approach is also tested on some PDPTW instances comparing to benchmark solution. The result shows that it is effective for cases where the customer locations are clustered. More appropriate decoding method for random-location instances should be further investigated. A new problem instance that considered all features of GVRP-MDPDR is generated to test the proposed methodology. As the preliminary results shown, it is clear that there is still room for improvement. However, the proposed methodology will be useful for the practitioners as it includes many features regarding practical aspects. The improvement in route construction algorithm and decoding method which is suitable for various forms of geographical data should be further investigated.

Acknowledgment

This work was supported by Thailand Research Fund under Grant MLSC525009.

References

[1] G. Nagy and S. Salhi. "Heuristics algorithm for single and multiple depot vehicle routing problem with pickups and deliveries," in *European Journal of Operational Research*, Vol. 162, 2005, pp. 126-141.

[2] G. Berbeglia, J.-F. Cordeau, and G. Laporte. "Dynamic pickup and delivery problems," in *European Journal of Operational Research*, 2009 In Press, Corrected Proof, DOI: 10.1016/j.ejor.2009.04.024.

[3] W.P. Nanry, J.W. Barbes.(2000) "Solving the pickup and delivery problem with time windows using reactive tabu search," in *Transportation Research B* Vol. 34, 2000, pp. 107-121.

[4] J. Renaud, F.F. Boctor, J. Ouenniche (2000). "A heuristics for the pickup and delivery traveling salesman problem.," in *Computer & Operation Research* Vol. 27, 2000, pp. 905-916.

[5] S. Ropke, D. Pisinger. "An adaptive large neighborhood search heuristic for the pickup and delivery with time windows," in *Transportation Science*, Vol. 40, 2006, pp. 455-472. Available : http://www.diku.dk/hjemmesider/ansatte/sropke/

[6] T. J. Ai and V. Kachitvichyanukul. "Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem," in *Computer & Industrial Engineering* Vol. 56, No. 1, 2009, pp. 380-387.

[7] T. J. Ai and V. Kachitvichyanukul. "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery" in *Computers & Operations Research*, Vol. 36 , 2009, pp. 1693-1702.

[8] T. J. Ai and V. Kachitvichyanukul. "A particle swarm optimization for vehicle routing problem with time windows," in *International Journal Operation Research.* Vol. 9, No. 4, pp.519-537.

[9] J. Kennedy and R Eberhart. "Particle Swarm Optimization," in *Proceedings of IEEE international Conference on Neural Networks*, Volume 4, 1995, pp. 1942-1948.

[10] P. Pongchairerks and V. Kachitvichyanukul. " A non-homogenous particle swarm optimization with multiple social structures," in *Proceedings of International Conference on Simulation and Modeling*, 2005. (Paper A5-02).

[11] P. Pongchairerks and V. Kachitvichyanukul. "Particle swarm optimization algorithm with multiple social learning structures," in *International Journal Operational Research*, Vol.6, No. 2, pp. 176-194.

[12] H. Li and A. Lim. "Pickup and delivery problem with time windows Instances,"2001. Available : http://www.sintef.no/Projectweb/TOP/Problems/PDPTW/Li--Lim-benchmark/

[13] M.M. Solomon. "Vehicle Routing Problem with Time windows Benchmarks Problems,"1987. Available : http://web.cba.neu.edu/~msolomon/problems.htm

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)