Hybrid Particle Swarm Optimization for Vehicle Routing Problem with Time Windows

S.MASROM¹, SITI.Z.Z. ABIDIN², A.M. NASIR³
Faculty of Computer and Mathematical Science
Universiti Teknologi MARA, MALAYSIA
¹suray078@perak.uitm.edu.my
²sitizaleha533@salam.uitm.edu.my
³arfah485@perak.uitm.edu.my

A.S.ABD. RAHMAN
Computer and Information Science Department
Universiti Teknologi PETRONAS, MALAYSIA
sanirahman@petronas.com.my

Abstract: Vehicle routing problem with Time Window (VRPTW) has received much attention by researchers in solving many scheduling applications for transportation and logistics. The objective of VRPTW is to use a fleet of vehicles with specific capacity to serve a number of customers with various demands and time window constraints. As a non-polynomial (NP) hard problem, the VRPTW is complex and time consuming, especially when it involves a large number of customers and constraints. This paper presents a hybrid approach between Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) for solving VRPTW. The reason for hybridization is to overcome the problem of premature convergence that exists in standard PSO. Premature convergence often yields partially optimized solutions because of particles stagnation. The proposed hybrid PSO implements a mechanism that automatically trigger swarm condition which will liberate particles from sub-optimal solutions hence enabling progress toward the maximum best solution. A computational experiment has been carried out by running the hybrid PSO with the VRPTW benchmark data set. The results indicate that the algorithm can produce some improvement when compared to the original PSO.

Key-Words: Vehicle Routing Problem with Time Windows, Particle Swarm Optimization, Genetic Algorithm, Mutation, Hybridization, Solution quality.

1 Introduction
The vehicle routing problem (VRP) was originally introduced by Dantziq and Ramser in 1959[1]. It is a problem to design a set of vehicle routes in which a fixed fleet of delivery vehicles from one or several depots to a number of customers have to be set with some constraints. To this date of literature, many kinds of VRP model have been introduced namely VRP with multiple depot, VRP with pickup and delivery, VRP with backhauls, capacitated VRP and VRP with time window (VRPTW).

The VRPTW encountered very frequently in making decision about the distribution of goods and services. The problems concern with minimizing the total cost of delivery for various demands but it is must be within specific time window constraint before the vehicle returning to depot. It has been considered that VRPTW is a non-polynomial (NP) hard problem, which means that the process of finding the optimal solution is considered as exponential to the problem size. This translates into significant cost (time) to the performance of the algorithm. As a consequence, a variation of metaheuristics based algorithms has been applied to VRPTW including Particle Swarm Optimization (PSO).

The concept of PSO is easy to be implemented and it involves only a few parameters to be adjusted. However, in some cases, PSO have a drawback in promising good optimal solutions[2]. This is due to the lack of diversity nature in the PSO which reduces its exploration capability in searching for wide vector of solutions. In order to increase the diversity nature in PSO, many researchers have incorporated mutation operator from the GA concepts into the PSO algorithms [3-4]. The results of the hybridization techniques are proven to be superior to the single PSO.

The remaining content of this paper is organized as follows. Section 2 presents the related works on PSO for VRP. Section 3 describes the VRPTW concept. The explanation on PSO, GA and the proposed hybrid PSO are given in Section 4, 5 and 6 respectively. Then, the computational test and results are given in Section 7. Lastly, Section 8 concludes the paper with a short summary.
2 Related works

Based on the literature, it can be seen that most of the metaheristics techniques on VRP were concentrated on algorithms hybridization rather than the single algorithm implementation [5-7]. The same interest also has been given to the PSO algorithm. A number of works have shown significant improvements to the VRP optimal solutions with the implementation of PSO hybridization. As for example, a combination of PSO with multiple phase neighborhood search and greedy randomized adaptive search (MPNSGRASP) has been effectively solved VRP problem[8]. The purpose this hybridization is for expanding the neighborhood strategy in PSO. In another work, PSO has also been hybridized with ACO for performance improvement [9]. This technique is proven to be effective when was implemented on VRP for grain logistic. A method that combines PSO with local search and simulated annealing is another technique for PSO hybridization. The computational experiments have showed that the proposed algorithm is feasible and effective for capacitated vehicle routing problem, especially for large scale problems [6].

3 Vehicle Routing Problem with Time Window (VRPTW)

The VRPTW is given by a fleet of homogeneous vehicles, \( V \) that is limited to a capacity, \( Q \). In a graph theoretical perspective, VRPTW can be defined as a complete undirected graph \( G=(N,E) \) with a vertex set of \( N \) and an edge set of \( E \). Each vertex, \( v_i (i=1..n) \), represents a customer while \( v_0 \) is corresponding to the depot. The set of edges represent the link between the depot and customers and customers to customers. The arcs are represented by \( E=\{(v_i,v_j): i\neq j\} \) and the distances, \( d_{ij} \), for each arc is calculated with Euclidean computations.

The objective of the VRPTW is to design a set of minimal cost routes, one set of customers for each vehicle in such that:
- Each customer is serviced exactly once;
- Every route originates and terminates at depot, \( v_0 \);
- The time window and capacity constraint are restricted to each vehicle.

The mathematical model for the VRPTW is based on the model found in [10].

4 Particle Swarm Optimization (PSO)

The basic PSO was proposed by Kennedy and Ebehart in 1995[11]. It is a population based metaheuristics that mimic the cognitive and social behaviors of birds flocking and fish schooling. The populations or solutions in PSO are called as particles which ‘flying’ through a problem’s search space and always accelerating towards better solutions.

The particles consist of a D-dimensional position vector \( \vec{x} \), and a D-dimensional velocity vector \( \vec{v} \). The \( i \)th particle position can be represented as \( \vec{x} = [x_{i1}, x_{i2}, x_{i3}, ..., x_{iD}] \) while the \( i \)th particle velocity can be indicated as \( \vec{v} = [v_{i1}, v_{i2}, v_{i3}, ..., v_{iD}] \). Fig. 1 illustrates the PSO algorithm.

![Fig. 1: PSO algorithm](image)

The algorithm starts by randomly initialize a population of particles. Then, it is searching for good optima solutions by evaluating and updating the current population at every time of generation or iteration.

In population generation of each iteration \( t \), every velocity of each particle \( (v_{it}) \) in the PSO has to move towards the best fitness based on the information of previous experience. The information consist of previous velocity, \( (v_{it}) \), cognitive perception of each particle, \( (c_1*rand*(pbest-x_{it}))* \) and social interaction among them, \( (c_2*rand*(gbest – x_{it}))* \). The formula for the velocity update is:

\[
v_{it} = v_{it-1} + c_1*rand*(pbest-x_{it-1}) + c_2*rand*(gbest-x_{it-1})
\]
The acceleration coefficient factors, $c_1$ and $c_2$, are two positive constants which have been experimentally proved to be effective with a value in a range of $2.0 – 2.5$ [12]. The $r_1$ and $r_2$ are two different generated random numbers in the uniform range of $[0:1]$.

In this research, the focus is on *inertia weight* PSO introduced by Shi and Eberhart [13]. The inertia factor, ($w$), is used to control the amount of a particle’s previous velocity. Its can be represented as the following formula:

$$v_{it} = w v_{i(t-1)} + c_1 r_1 (p_{best} - x_{i(t-1)}) + c_2 r_2 (g_{best} - x_{i(t-1)})$$

After updating each particle velocity, the position for each particle is updated by adding current position to the new update velocity. The expression is $x_{it} = x_{i(t-1)} + v_{it}$.

### 5 Genetic Algorithm (GA)

The GA is an optimization algorithm that simulates the evolution of creatures. In GA, the solutions in a state space are expressed as individuals and every individual is composed of chromosomes. A set of individual is called a GA population. Similar to PSO, the initial population for GA is randomly generated. However, the GA performs population reproduction based on selection, recombination and mutation. Fig. 2 shows the algorithm for GA.

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions are typically more likely to be selected.

The next process is crossover operation. It is a process that combines two chromosomes or parents to produce a new chromosome or offspring. The idea behind crossover is to generate a new chromosome that should be better than both of the parents.

Mutation is a genetic operator that alters one or more gene values in a chromosome from its current state. With the new gene values, the genetic algorithm may be able to arrive at a better solution than was previously possible. Mutation is an important part of the genetic search as it helps to prevent the population from stagnating at any local optima.

### 6 Hybrid PSO

This paper introduced a hybrid version of PSO (HPSO) for solving the VRPTW. Figure 3 is the flowchart of the algorithm.

![Fig. 3: IPSO algorithm for VRPTW](image-url)
The new procedures in HPSO are VRP solution representations, decoding method, particles health detection and mutation operation.

6.1 Solution representations
In the HPSO, the method proposed by [14] has been selected for VRP solutions in particle representation. The method divides every particle position into \((n+2m)\) dimension and each dimension is encoded as a real number. The first \(n\) dimensions are representing the numbers of customer and the last \(2m\) dimensions are related to number of vehicle with its reference point in Cartesian map. The reference point is represented by two dimensions for each vehicle. Fig. 4 shows the solution representation applied in HPSO.

\[
\begin{array}{ccccccc}
(1) & 0.5 & (2) & 0.9 & (3) & 0.4 & (4) \\
(5) & 0.8 & (6) & 0.2 & (7) & 0.1 & (8) \\
\end{array}
\]

\[
\begin{array}{cccc}
(5) & 1.1 & (6) & 1.6 \\
(7) & 1.4 & (8) & 1.0 \\
\end{array}
\]

\(n = 8\) customers
\(m = 2\) vehicles
\(2m = 4\) dimensions

Fig. 4: solution representation for each particle

6.2 Decoding method
The steps of decoding method in HPSO are:
- Construct customer priority list based on the position value. It can be done by sorting the first \(n\) dimensional values of particle in an ascending order. Then, arrange the customer index according to the position value. Fig. 5 shows the process of customer priority list construction.
- Extract the vehicle reference point from the last \(2m\) dimensions of particle. Construct vehicle priority matrix based on the relative distance between the point and customer location (geographical coordinate). The closest customer will be listed as the most prioritized to be served.
- Finally, construct vehicles routes based on customer priority list and vehicle priority matrix. Fig. 6 illustrates the steps of vehicle routes construction.

\[
\begin{array}{cccccc}
(1) & 0.5 & (2) & 0.9 & (3) & 0.4 \\
(4) & 0.3 & (5) & 0.8 & (6) & 0.2 \\
(7) & 0.1 & (8) & 0.7 \\
\end{array}
\]

\[
\begin{array}{cccc}
(7) & 0.1 & (6) & 0.2 \\
(4) & 0.3 & (3) & 0.4 \\
(1) & 0.5 & (8) & 0.7 \\
(5) & 0.8 & (2) & 0.9 \\
\end{array}
\]

Fig 5: Construction of customer priority list

<table>
<thead>
<tr>
<th>Vehicle no</th>
<th>Reference point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

Vehicle priority matrix

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>0-4-1-8-5-2-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle 1</td>
<td>0-7-6-3-0</td>
</tr>
</tbody>
</table>

Fig 6: Construction of vehicle routes

Each of vehicle routes starts and ends the journey at a depot as represented by 0 in the route list. It is to comply with the VRPTW constraint.

6.3 Detect particle health
The idea behind particle health detection was adapted from the PSO Triggered Mutation algorithm introduced by [15]. A particle is considered healthy if it has found a new personal best position at the current iteration. The number of particles with personal best improvement will be identified from the swarm population. When a certain percentage of particles in the swarm population are not healthy, the population condition can be considered as unhealthy and not stronger. In this condition, the particles have fallen into local optima and stagnated.

Every time of iteration, the occurrence of unhealthy condition is counted. If the number
counted is bigger than certain number of iterations, the swarm is ready for mutation process.

6.4 Mutation
The purpose of mutation operation is to slightly change the value of particles’ positions so that it will increase the possibility of many potential solutions to be explored. A variety type of mutation operations have been proposed for PSO algorithm[3]. However, the operations are not really suitable and not specific for VRPTW solutions. In VRPTW solutions, the customer priority position is hardly dependent on the initialization position. So, mutating at random particle position will change the customer priority arrangement. It will cause to result inferiority to solutions. So, in the HPSO, the mutation is done on the vehicle location or reference point. The formula for mutation operation is \( X_{\text{new}} = X_{\text{old}} + \text{gaussian}(p) \). The gaussian function returns a random number drawn from a Gaussian distribution with a standard deviation of \( p \). The value of \( p \) is 0.1 times the length of dynamic range of the particle dimension [4]. This is to ensure the vehicle reference point for each vehicle is within the search vector.

7 Experiments and Results
Computational experiments have been conducted to evaluate the performance of the HPSO using Solomon benchmark data set [10]. The algorithm is implemented in MATLAB 2010 on a personal computer with Intel Pentium 1.20 GHz-3GBRAM and running under Windows 7. A total of 20 runs for each case were conducted and an average result is obtained. The parameter configuration is listed in Table 1.

<table>
<thead>
<tr>
<th>Data set</th>
<th>C1</th>
<th>C2</th>
<th>R1</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling horizon</td>
<td>Clustered</td>
<td>Long (more than 30 routes)</td>
<td>Short (5-10 routes)</td>
<td>Short (5-10 routes)</td>
</tr>
<tr>
<td>Vehicle capacity</td>
<td>200</td>
<td>700</td>
<td>200</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 2: Data set

7.1 Benchmark data set
The Solomon’s benchmark [10] data set have been used for the experiment. It is comprised of variety of problem situations such as customer location, vehicles information, customers’ demand and corresponding time windows. Table 2 lists the data set used in the experiment.

7.2 Results
The solution quality for HPSO is slightly better than original PSO for clustered data set (C2) with long routes for the case of 25 numbers of customers. However, no improvement can be seen on the short routes (C1). At the random (R) and semi random (RC1) distribution, the HPSO has successfully produced better results than the PSO. Table 3 shows the results for VRPTW with 25 customers.

Table 4 has showed that all instances of problem with 100 customers have gained some improvement to the optimal solutions from the HPSO implementation. An apparent achievement can be seen from the random distribution (R1) customers.

Table 3: Computational results for 25 customers

Table 4: Computational results for 100 customers

Table 1: Parameter setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of particle</td>
<td>100</td>
</tr>
<tr>
<td>Number of iteration</td>
<td>1000</td>
</tr>
<tr>
<td>Inertia weight, ( w )</td>
<td>Linear decreasing from 0.9 to 0.4</td>
</tr>
<tr>
<td>( X_{\text{min}}, V_{\text{max}} )</td>
<td>0, 100</td>
</tr>
<tr>
<td>Personal acceleration, ( c_1 )</td>
<td>2</td>
</tr>
<tr>
<td>Global acceleration, ( c_2 )</td>
<td>2</td>
</tr>
<tr>
<td>Healthy percentage</td>
<td>60%</td>
</tr>
<tr>
<td>Unhealthy occurrence</td>
<td>3</td>
</tr>
<tr>
<td>mutation rate</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Although the time taken for getting the last optimal solutions in HPSO is a little longer than the PSO, the range is still within reasonable limits. Table 5 is the comparison of computational time for PSO and HPSO with 25 and 100 customers.

<table>
<thead>
<tr>
<th>Prob.</th>
<th>25 Customers</th>
<th>100 Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSO</td>
<td>HPSO</td>
</tr>
<tr>
<td>C101</td>
<td>36</td>
<td>41</td>
</tr>
<tr>
<td>C102</td>
<td>35</td>
<td>38</td>
</tr>
<tr>
<td>C201</td>
<td>59</td>
<td>62</td>
</tr>
<tr>
<td>C202</td>
<td>57</td>
<td>63</td>
</tr>
<tr>
<td>R101</td>
<td>34</td>
<td>40</td>
</tr>
<tr>
<td>R102</td>
<td>35</td>
<td>40</td>
</tr>
<tr>
<td>RC101</td>
<td>38</td>
<td>45</td>
</tr>
<tr>
<td>RC102</td>
<td>38</td>
<td>46</td>
</tr>
</tbody>
</table>

8 Conclusions
This paper presents a method for solving VRPTW by using hybrid PSO and GA algorithm. From the empirical results, it has been revealed that the hybrid method proves to be more effective than the original PSO in many cases of VRPTW especially with the high number and randomly distributed customers. On the other hand, this hybrid algorithm requires more computational time as compared to the original PSO. In future work, the effectiveness of the proposed hybrid PSO should be tested on other types of VRP problems.

9 Acknowledgement
The authors would like to thank Kementerian Pengajian Tinggi MALAYSIA for its financial support to this project.

References: