An Improved Particle Swarm Optimization Algorithm for Solving Vehicle Routing Problem with Simultaneous Pickup and Delivery

Rong Wei¹, Tongliang Zhang ², Hui Tang ²

¹ College of Science, Hebei polytechnic university, Tangshan, Hebei 063009, P.R. China;
² Research Institutes of Highway, Ministry of Transport, Beijing 100088, P.R. China;
{weirong@heut.edu.cn, tl.zhang@rioh.cn}

Abstract. Vehicle routing problem with simultaneous pickup and delivery (VRPSPD) is becoming research hot field for more and more instances in logistics practical. This paper proposed an improving particle swarm optimization algorithm based on multiple social structures for solving VRPSPD. The decoding of particle consists two parts: the first part consist m dimensional for m customers, and the second part presentation of vehicle route orientation which consist 2n dimensions for n vehicles. The particle is transformed to a customers list and a vehicles matrix. A benchmark dataset is used to validate the performance of proposed algorithm. Comparing with prior works, promising results indicate that the proposed approach may hold high potentials for generating powerful tool for solving VRPSPD and their other attributes.

Keywords: Vehicle routing problem, simultaneous pickup and delivery, particle swarm optimization, multiple social structures, logistics

1 Introduction

The Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) is a variant of the Capacitated Vehicle Routing Problem (CVRP), in which clients require both pickup and delivery services. In some particle customers may have both a delivery and a pick-up demand, such as soft drink industry where empty bottles must be returned. These customers may not accept to be serviced separately for the delivery
and pick-up they require, because a handling effort is necessary for both activities and this effort may be considerably reduced by a simultaneous operation. Applications of the VRPSPD can be found especially within the Reverse Logistics context. Companies become interesting in gaining control over the whole lifecycle of products, especially when environmental issues are involved. Companies are increasingly faced with the task of managing the reverse flow of finished goods or raw-materials.

The VRPSPD is clearly NP-hard since it can be reduced to the CVRP when all the pickup demands are equal to zero. Min [1] first proposed the VRPSPD in 1989, which have a gap of more than 10 years without any works published. Recently, the studies on VRPSPD increase rapidly for the development of reverse logistics and retail distribution. Dethloff [2] proposed a mathematical formulation for VRPSPD from the point of view of reverse logistics to minimize the total traveled distance subject to maximum capacity constraint of the vehicle. The development of modern heuristics provides strong techniques for resolving the problem of multi-objective optimization. Recently, a few heuristics algorithms have been proposed for VRPSPD, like local search heuristics [3], tabu search algorithm[4], genetic algorithm [5,6]. Angelelli and Mansini developed an exact algorithm based branch-and-price for VRPSPD with time windows [7]. Particle swarm optimization (PSO) is a population-based stochastic optimization technique developed by Kennedy and Eberhart [8], which motivated by the group organism behavior like bird flock and fish shoal. The application PSO on VRPSPD is still rare. One is the work of Jin and Kachitvichyanukul [9], where they proposed the random key-based solution representation and decoding method for implementing PSO for VRPSPD. In this paper, we developed mathematical formulation for VRPSPD, and further proposed an improved PSO algorithm to solve it.

The remainder of the paper is organized as follow: section 2 is material and methods, where mathematical formulation and proposed algorithm of VRPSPD are included. Section 3 discusses the experiment results on a benchmark dataset. Finally, Section 4 concludes the research.

2 Material and Methods
2.1 VPRSPD mathematics formulation

The VPRSPD can be formulated as a graph \( G = (V,A) \), where \( V = \{v_0, \ldots, v_n\} \) is a vertex set, and \( A = \{(v_i,v_j)\} \) is arc set. Distance matrix \((d_{ij})\) and time matrix \((t_{ij})\) is the parameters of arc \( a_{ij} = (v_i,v_j) \). Vertex \( v_0 \) is the depot at which \( n \) vehicles are stationed. Each customer corresponds vertex \( v_i \) and has a non-negative pickup quantity \( p_i \), delivery quantity \( q_i \). The vehicle has some fix parameters, such fixed cost of \( f_i \), capacity \( Q \), and service duration limit \( D \), the vehicle routes in VPRSPD must meet the restriction: (1) the total routing cost is minimized; (2) the total duration of each route does not exceed the limit \( D \) of the servicing vehicle; (3) the total vehicle load in any route does not exceed the capacity \( Q \) of the vehicle. (4) each vertex (customer) is visited exactly once by one vehicle; and (5) each vehicle starts from depot \( v_0 \) and return to it.

The mathematics formulation of VPRSPD is presented below:

\[
\text{Min} \text{ } C = \sum_{i=1}^{n} \sum_{j=1}^{m} C_{ijk} x_{ijk} + g \sum_{i=0}^{n} \sum_{j=1}^{m} \sum_{k=1}^{n} \sum_{i=1}^{m} d_{ij} x_{ijk} \tag{1}
\]

where, \( x_{ijk} \) indicates whether the arc \((i,j)\) is traversed by vehicle \( k \).

\[
x_{ijk} = \begin{cases} 
0 & \text{if vehicle } k \text{ does not traversed arc } (i,j) \\
1 & \text{if vehicle } k \text{ traversed arc } (i,j) 
\end{cases} \tag{2}
\]

Subject to,

\[
\sum_{j=1}^{m} x_{ijk} = 1 \quad \forall \ i, k \tag{3}
\]

\[
\sum_{i=1}^{n} x_{ijk} = \sum_{j=1}^{m} x_{ijk}, \text{ for } 1 \leq i \leq m, 1 \leq k \leq n \tag{4}
\]

\[
y_{ijk} \leq x_{ijk} Q, \text{ for } 0 \leq i \leq m, 1 \leq j \leq m + 1, 1 \leq k \leq m \tag{5}
\]

\[
\sum_{j=1}^{m} y_{ijk} = \sum_{j=1}^{m} q_i \sum_{k=1}^{n} x_{ijk}, \text{ for } 1 \leq k \leq n \tag{6}
\]

where, \( y_{ijk} \) presents the load of vehicle \( k \) which traverses arc \((i,j)\).
The objective function Eq. (1) indicates that this model minimizes routing cost, which consists of transportation fixed cost and variable cost. Eq. (3) and (4) represent that every customer is visited by exactly one vehicle. Vehicle load constraints are explained in (5)-(7). Constraint (5) states that if vehicle \( k \) serving customer \( j \) after serving customer \( i \) (\( x_{ijk} = 1 \)), the corresponding load (\( y_{ijk} \)) must at most equal to the vehicle load capacity (\( Q \)); and otherwise the load \( y_{ijk} = 0 \) if \( x_{ijk} = 0 \). Constraint (6) assures that all customer deliveries are from the depot. It states that the load of a vehicle at the departure from the depot must be equal to the total load for customer deliveries of the corresponding vehicle. Constraint (7) balances the load of a vehicle after it serves a customer.

\[
\sum_{j=1}^{m} y_{ij} + (p_j - q_j) \sum_{k=1}^{n} y_{ik} = \sum_{m=1}^{M} y_{iq}, \text{ for } 1 \leq j \leq m, 1 \leq k \leq n
\]  

(7)

The objective function Eq. (1) indicates that this model minimizes routing cost, which consists of transportation fixed cost and variable cost. Eq. (3) and (4) represent that every customer is visited by exactly one vehicle. Vehicle load constraints are explained in (5)-(7). Constraint (5) states that if vehicle \( k \) serving customer \( j \) after serving customer \( i \) (\( x_{ijk} = 1 \)), the corresponding load (\( y_{ijk} \)) must at most equal to the vehicle load capacity (\( Q \)); and otherwise the load \( y_{ijk} = 0 \) if \( x_{ijk} = 0 \). Constraint (6) assures that all customer deliveries are from the depot. It states that the load of a vehicle at the departure from the depot must be equal to the total load for customer deliveries of the corresponding vehicle. Constraint (7) balances the load of a vehicle after it serves a customer.

2.2 Particle Swarm Optimization algorithm for VPRSPD

PSO is a population-based optimization algorithm. The detail of PSO algorithm can refer to the literature [10]. Here, we use an improved PSO algorithm with multiple social learning structures[11] to solve VPRSPD. Except for the global best (gbest) and particle best (pbest), other two parameters are involved, the local best (lbest). The local best is the best position of among several adjacent particles.

The decoding of particle in swarm is the key elements for effective implementation of PSO for VRPSPD. It consists two parts: the first part is \( m \) dimensions related to \( m \) customers, the second part is presentation of vehicle route orientation. Route orientation of a vehicle is defined as a point that represents vehicle service area. A route orientation point has its coordinate (\( x, y \)). Hence, the presentation of vehicle route orientation will consist \( 2n \) dimensions of a particle for \( n \) vehicles.

The details of the PSO algorithm for solving VRPSPD is illuminated in Table 1, and the notation of parameters in algorithm are listed in Table 2.

<table>
<thead>
<tr>
<th>Table 1. Proposed PSO algorithm for VRPSPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>01. Initialize swarm ( S ) with ( n ) particles, initialize PSO parameters</td>
</tr>
<tr>
<td>02. For each particle ( i ) in ( S )</td>
</tr>
<tr>
<td>03. Initialize random position ( P_i \in [\theta_{\text{min}}, \theta_{\text{max}}] )</td>
</tr>
</tbody>
</table>
04. Velocity \( v_i = 0 \)
05. Initialize pbest with a copy of the position for particle, \( P_i^{pbest} = P_i \)

06. **End For**

07. Set iteration \( \tau = 1 \)

08. decode \( i \)-th particles in the \( \tau - \text{th} \) iteration \( P_i(\tau) \) to a set of vehicle route \( R_i \).

09. **While** the termination conditions are not met

For \( i = 1 \ldots L \)

10. Compute the fitness value \( f(P_i) \)

11. Get the global best (gbest)

12. Get the local best in \( i \)-th particle

13. Update pbest, If \( f(P_i) < f(pbest_i) \), Then \( pbest_i = P_i \)

14. Update gbest, If \( f(P_i) < f(gbest) \), Then \( gbest = P_i \)

15. Update lbest, If \( f(P_i) < f(lbest_i) \), Then \( lbest_i = P_i \)

16. Update the velocity and the position of \( i \)-th particle

17. Update Inertia weight \( w(\tau) = w(T) + \frac{T-\tau}{T-1}(w(1) - w(T)) \)

For \( h = 1 \ldots H \)

18. \( v_a(\tau + 1) = w(\tau)v_a(\tau + 1) + C_p u(pbest_a - p_a(\tau)) + C_s u(gbest_a - p_a(\tau)) \)

19. \( + C_i u(lbest_a - p_a(\tau)) \)

20. \( P_a(\tau + 1) = P_a(\tau) + v_a(\tau + 1) \)

21. **If** \( P_a(\tau + 1) > \theta^{\text{max}} \) **Then**

22. \( P_a(\tau + 1) = \theta^{\text{max}} \),

23. \( v_a(\tau + 1) = 0 \)

**End If**

24. **End If**

25. **If** \( P_a(\tau + 1) < \theta^{\text{min}} \) **Then**

26. \( P_a(\tau + 1) = \theta^{\text{min}} \),

27. \( v_a(\tau + 1) = 0 \)

**End If**

28. **End If**

29. **End For**

30. **End For**

31. \( \tau = \tau + 1 \)

32. **End While**
### Table 2. The notation of parameters in proposed algorithm

<table>
<thead>
<tr>
<th>Name</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>iteration index; $\tau = 1...T$</td>
</tr>
<tr>
<td>$i$</td>
<td>particle index; $i = 1...L$</td>
</tr>
<tr>
<td>$h$</td>
<td>dimension index; $h = 1...H$</td>
</tr>
<tr>
<td>pbest</td>
<td>Personal best in $i$-th particle</td>
</tr>
<tr>
<td>gbest</td>
<td>Global best in swarm</td>
</tr>
<tr>
<td>lbest</td>
<td>Local best position in $i$-th particle, among all pbest from K neighbors of the $i$-th particle, set the personal best which obtains the least fitness value to be lbest.</td>
</tr>
<tr>
<td>$w(\tau)$</td>
<td>Inertia weight in the $\tau$-th iteration</td>
</tr>
<tr>
<td>$w(T)$</td>
<td>Last Inertia weight</td>
</tr>
<tr>
<td>$v_{ih}(\tau)$</td>
<td>velocity of $i$-th particle at the $h$-th dimension in the $\tau$-th iteration</td>
</tr>
<tr>
<td>$P_{ih}(\tau)$</td>
<td>position of the $i$-th particle at the $h$-th dimension in the $\tau$-th iteration</td>
</tr>
<tr>
<td>$C_p$</td>
<td>personal best position acceleration constant</td>
</tr>
<tr>
<td>$C_g$</td>
<td>global best position acceleration constant</td>
</tr>
<tr>
<td>$C_l$</td>
<td>local best position acceleration constant</td>
</tr>
<tr>
<td>$u$</td>
<td>uniform random number in the interval $[0, 1]$</td>
</tr>
<tr>
<td>$\theta_{\text{max}}$</td>
<td>maximum position value</td>
</tr>
<tr>
<td>$\theta_{\text{min}}$</td>
<td>minimum position value</td>
</tr>
</tbody>
</table>

### 3 Results and discussion

The proposed algorithm is validated on the benchmark data set of Delthloff [2], which comprises four data sets named SCA3, SCA8, CoN3 and CON8, respectively. The same data sets are also used in prior work for performance measurement. In order to compare with prior work in same scenario, the problem parameters in VRPSDP formulation are set as follow: the fixed cost per vehicle $f=0$; service duration limit $D = \infty$; variable cost per distance unit $g=1$.

The algorithm is implemented in matlab on PC with 2.6G, 2G RAM. The PSO parameters are set based on the result of some preliminary experiments, and the parameters are listed in Table 3.
Table 3. PSO parameters in VRPSDP

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particel Number</td>
<td>40</td>
</tr>
<tr>
<td>Neighbor number</td>
<td>8</td>
</tr>
<tr>
<td>Maximum iteration number</td>
<td>500</td>
</tr>
<tr>
<td>First inertia weight</td>
<td>( w(1)=0.95 )</td>
</tr>
<tr>
<td>Last inertia weight</td>
<td>( w(T)=0.45 )</td>
</tr>
<tr>
<td>Pbest position acceleration constant</td>
<td>( C_p=1 )</td>
</tr>
<tr>
<td>Gbest position acceleration constant</td>
<td>( C_g=0.1 )</td>
</tr>
<tr>
<td>Lbest position acceleration constant</td>
<td>( C_l=0.9 )</td>
</tr>
</tbody>
</table>

The proposed algorithm implements 10 replications on each data set. The best results among 10 PSO iterations are listed in Table 4, where for facilitating comparison, the results by other methods on the same dataset are also listed. It is shown that the proposed method is encouraging. Our results is better than that of Dethloff[2] and The Jin Ai and Voratas [9] for all data sets, better than Tang and Galvao[4] for SCA8 data set, and better than Bianchessi and Righini[12] for CON3 data set.

Table 4. The total cost obtained by different algorithms on four data set

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SCA3</td>
<td>746.6</td>
<td>674.2</td>
<td>684.6</td>
<td>675.8</td>
<td>672.8</td>
<td></td>
</tr>
<tr>
<td>SCA8</td>
<td>1166.4</td>
<td>1044.4</td>
<td>1035.7</td>
<td>1041.8</td>
<td>1039.4</td>
<td></td>
</tr>
<tr>
<td>CON3</td>
<td>597.3</td>
<td>564.2</td>
<td>568.5</td>
<td>569.6</td>
<td>565.3</td>
<td></td>
</tr>
<tr>
<td>CON8</td>
<td>860.6</td>
<td>774.3</td>
<td>776.4</td>
<td>798.3</td>
<td>767.9</td>
<td></td>
</tr>
</tbody>
</table>

4 Conclusions

In this paper we developed an improved PSO algorithm for solving VPRSPD. The concept of local best is considered in PSO, which presents the best position of among several adjacent particles of \( i \)-th particle. It strengthens the particles learning in
multiple social learning structures. The decoding method of PSO for VPRSPD is presented. Each particle in swarm consist \((m+2n)\) dimensions corresponding to \(m\) customers and \(n\) vehicles. A benchmark dataset is used to validate the performance of proposed algorithm. Compared to prior works, the results of proposed algorithm is encouraging.

**Acknowledgments.** This work was supported financially in part by 863 Foundation (Grant No. 2006AA04A105) and Tangshan Science Foundation (Grant No. 09110219C).

**References**