Volume 36
Issue 5
May 2009
ISSN 0305-0548
ELSEVIER
computers & operations research
Part Special Issue:
Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X)
Guest Editors:
Juan A. Mesa, Blas Pelegrin and Justo Puerto
Editor: Stefan Nickel
Available online at
ScienceDirect
www.sciencedirect.com

Computers & Operations Research 36 (2009) 1693 – 1702

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery

The Jin Ai*, Voratas Kachitvichyanukul

Industrial Engineering and Management, School of Engineering and Technology, Asian Institute of Technology, P.O. Box 4, Klong Luang, Pathumtani 12120, Thailand

ARTICLE INFO

Available online 11 April 2008

Keywords:
Vehicle routing problem
Simultaneously pickup and delivery
Particle swarm optimization
Random key representation
Metaheuristics

ABSTRACT

This paper proposes a formulation of the vehicle routing problem with simultaneous pickup and delivery (VRPSPD) and a particle swarm optimization (PSO) algorithm for solving it. The formulation is a generalization of three existing VRPSPD formulations. The main PSO algorithm is developed based on GLNPSO, a PSO algorithm with multiple social structures. A random key-based solution representation and decoding method is proposed for implementing PSO for VRPSPD. The solution representation for VRPSPD with n customers and m vehicles is a $(n+2m)$-dimensional particle. The decoding method starts by transforming the particle to a priority list of customers to enter the route and a priority matrix of vehicles to serve each customer. The vehicle routes are constructed based on the customer priority list and vehicle priority matrix. The proposed algorithm is tested using three benchmark data sets available from the literature. The computational result shows that the proposed method is competitive with other published results for solving VRPSPD. Some new best known solutions of the benchmark problem are also found by the proposed method.

Scope and Purpose

This paper applies a real-value version of particle swarm optimization (PSO) algorithm for solving the vehicle routing problem with simultaneous pickup and delivery (VRPSPD). The VRPSPD formulation is reformulated and generalized from three existing formulations in the literature. The purposes of this paper are to explain the mechanism of the PSO for solving VRPSPD and to demonstrate the effectiveness of the proposed method.

1. Introduction

The vehicle routing problem (VRP) is a generic name given to a class of problems to determine a set of vehicle routes, in which each vehicle departs from a given depot, serves a given set of customers, and returns back to the same depot. Various types of service appear in practical situations, while physical delivery of goods is the most common one.

The basic VRP involves a single depot, a fleet of identical vehicles that stations at the depot, and a set of customers who require delivery of goods from the depot. The objective of basic VRP is to minimize the total routing cost, subject to maximum working time and maximum capacity constraints on the vehicles [1]. Besides the basic VRP, many VRP variants may appear since there are many possibilities in real-life problem settings and characteristics, for example: the number of depots, type of vehicle, and customer requirements. Toth and Vigo [2] provide comprehensive details on VRP, its variants, formulation, and solution methods.

One extension of the basic VRP is the vehicle routing problem with simultaneous pickup and delivery (VRPSPD). In this variant, customers require not only the delivery of goods but also the simultaneous pick up of goods from them. A general assumption is that all delivered goods originate from the depot and all pickup goods must be transported back to the depot. Min [3] was inspired by a distribution problem of a public library and first introduced this extension as the VRPSPD for minimizing the total travel time of the route by considering the vehicle capacity as the problem constraint. His proposed solution procedure for the problem consists of three phases: clustering customer nodes, assigning vehicles to clusters, and creating the route of each vehicle.

After Min [3], some researchers also contributed on the mathematical formulation of VRPSPD and the solution techniques. Dethloff [4] discussed the importance of VRPSPD in the reverse logistic

* Corresponding author. Tel.: +66 252 46604; fax: +66 252 45697.
E-mail addresses: thejin.ai@ait.ac.th, jinai@mail.uajy.ac.id (T.J. Ai), voratas@ait.ac.th (V. Kachitvichyanukul).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.04.003

operations. He proposed a mathematical formulation for the problem to minimize the total traveled distance subject to maximum capacity constraint of the vehicle. He also developed an insertion-based heuristic that use four different criteria to solve the problem.

Salhi and Nagy [5] proposed four insertion-based heuristics for generating solution for VRPSPD. The basic steps of these heuristics are constructing partial routes for a set of customers, and then inserting the remaining customers to the existing route. These heuristic rules were differentiated mainly by the criteria for insertion and number of customers per insertion. Nagy and Salhi [6] also proposed a local search heuristic with four phases to solve VRPSPD. After finding an initial solution in the first phase, it is continuously improved in each of the following phases while maintaining a certain feasibility condition. In both papers, they addressed not only the VRPSPD, but also the mixed case of VRP where some customers require delivery and the other customers require pickup. They showed that the VRPSPD is a generalization of the mixed problem. In addition, they also extended the method for the multi depot case.

Tang and Galvao [7] developed a tabu search algorithm to solve VRPSPD. The algorithm combines several efforts to obtain alternative inter-route and intra-route solutions, includes relocation of a customer from one route to another route, interchange a pair of customers between two routes, crossover two routes, and 2-opt procedure. In their formulation, the VRPSPD is formulated to minimize the total traveled distance of the route subject to maximum distance and maximum capacity constraints on the vehicles.

Bianchessi and Righini [8] proposed heuristic algorithms for solving VRPSPD. Their work comprised of four different constructive algorithms, local search algorithms with various neighborhood structures, and tabu search algorithms. They were using Dethloff's VRPSPD formulation and their computational result outperformed result in Dethloff [4].

Dell'Amico et al. [9] was the first published work on exact method for solving VRPSPD. They presented an optimization algorithm based on column generation, dynamic programming, and branch and price method. However, the computational complexity of VRPSPD is evident from the computational result, in which 1 h of computational time sometimes is not enough for solving a small size problem consist of 40 customers.

It is noted that the VRPSPD can be seen as a pickup and delivery problem (PDP). In the recent classification on static pickup and delivery problem by Berbeglia et al. [10], the VRPSPD is called the multi-vehicle Hamiltonian one-to-many-to-one pickup and delivery problem with combined demands. By this definition, the problem consists of multi vehicles which its routes are a Hamiltonian cycle; the deliveries are from depot and the pickups will be transferred back to depot (one-to-many-to-one); the customer demand is combined which means that at least there is one customer with non zero pickup and delivery demand.

PSO is a population-based search method proposed by Kennedy and Eberhart [11], which motivated by the group organism behavior such as bee swarm, fish school, and bird flock. PSO imitated the physical movements of the individuals in the swarm as a searching method. A brief and complete survey on PSO mechanism, technique, and application is provided by Kennedy and Eberhart [12] and also Clerc [13]. While some other population-based search methods had been successfully applied in broader area of VRP, such as genetic algorithm [14–16] and ant colony optimization [17,18], the application of PSO on VRP is still rare. One is the work of Chen et al. [19], where the discrete version of PSO is combined with Simulated Annealing (SA) algorithm for solving the basic VRP.

There are two main contributions of this paper. First, it reformulates the VRPSPD as a direct extension of the basic VRP. As a result, the formulation of Min [3], Dethloff [4], Tang and Galvao [7] can be reduced to a special case of this reformulation. Second, it fills the gap of the application of PSO for VRP solution by showing how the real-valued version of PSO is applicable for solving VRPSPD.

The proposed algorithm in this paper is different from Chen's algorithm [19] in two aspects. First, the algorithm uses real value instead of discrete value for search variables. Second, it is implemented without the use of any local search method.

The remainder of this paper is organized as follows: Section 2 reviews the VRPSPD definition and mathematical formulation. Section 3 describes the proposed PSO algorithm for solving VRPSPD. Section 4 discusses the computational experiment of the proposed PSO on a benchmark data set. Finally, Section 5 concludes the result of this research and suggests further direction of the future research.

2. VRPSPD formulation

The VRPSPD can be formally defined as follows. Let $G=(V,A)$ be a graph where $V=\{v_0, v_1, \ldots, v_n\}$ is a vertex set, and $A=\{(v_i, v_j)|v_i, v_j \in V, i \neq j\}$ is an arc set. Associated with A are a distance matrix (d_{ij}) and a travel time matrix (t_{ij}). Vertex v_0 represents a depot at which m homogeneous vehicles are stationed, while the remaining vertices correspond to n customers. Each customer has a non-negative pickup quantity p_i, delivery quantity q_i, and a service time s_i. Every vehicle has a fixed cost of f, variable cost per distance unit g, capacity Q, and service duration limit D. The VRPSPD consists of designing a set of at most m routes such that

(1) each route starts and ends at the depot;
(2) each customer is visited exactly once by exactly one vehicle;
(3) the total vehicle load in any arc does not exceed the capacity of the vehicle assigned to it (Q);
(3) the total duration of each route (including travel and service times) does not exceed a preset limit D; and
(4) the total routing cost is minimized.

The mathematical formulation of VRPSPD is presented below following the preceding definition, which is a network flow-based formulation and a mixed integer linear program (MILP). The formulation is an extension of Christofides' basic VRP formulation [1].

Decision variables:

x_{ijk} a binary variable indicating whether arc (i,j) is traversed by vehicle k
$x_{ijk}=1$ if vehicle k traverses arc (i,j)
$x_{ijk}=0$ if vehicle k does not traverse arc (i,j)
y_{ijk} load of vehicle k while traverses arc (i,j)
δ_{ik} starting service time of customer i by vehicle k

Objective function

$$\text{Minimize } Z=f\sum_{k=1}^{m}\sum_{j=1}^{n}x_{0jk}+g\sum_{i=0}^{n}\sum_{j=1}^{n+1}\sum_{k=1}^{m}d_{ij}x_{ijk} \tag{1}$$

Subject to

$$\sum_{i=0}^{n}\sum_{k=1}^{m}x_{ijk}=1 \quad \text{for } 1\leqslant j\leqslant n \tag{2}$$

$$\sum_{j=0}^{n}x_{jik}=\sum_{j=1}^{n+1}x_{ijk} \quad \text{for } 1\leqslant i\leqslant n,\ \ 1\leqslant k\leqslant m \tag{3}$$

$$\sum_{j=1}^{n}x_{0jk}\leqslant 1 \quad \text{for } 1\leqslant k\leqslant m \tag{4}$$

$$\delta_{ik}+s_i+t_{ij}-\delta_{jk}\leqslant(1-x_{ijk})M \ \text{ for } 0\leqslant i\leqslant n,\ \ 1\leqslant j\leqslant n+1,\ \ 1\leqslant k\leqslant m \tag{5}$$

$$\delta_{n+1,k}-\delta_{0k}\leqslant D \quad \text{for } 1\leqslant k\leqslant m \tag{6}$$

$$y_{ijk} \leqslant x_{ijk} Q \quad \text{for } 0 \leqslant i \leqslant n,\ \ 1 \leqslant j \leqslant n+1,\ \ 1 \leqslant k \leqslant m \tag{7}$$

$$\sum_{j=1}^{n} y_{0jk} = \sum_{j=1}^{n} q_j \sum_{i=0}^{n} x_{ijk} \quad \text{for } 1 \leqslant k \leqslant m \tag{8}$$

$$\sum_{i=0}^{n} y_{ijk} + (p_j - q_j) \sum_{i=0}^{n} x_{ijk} = \sum_{i=1}^{n+1} y_{jik} \quad \text{for } 1 \leqslant j \leqslant n,\ \ 1 \leqslant k \leqslant m \tag{9}$$

$$x_{ijk} \in \{0, 1\} \quad \text{for } 0 \leqslant i \leqslant n,\ \ 1 \leqslant j \leqslant n+1,\ \ 1 \leqslant k \leqslant m \tag{10}$$

$$y_{ijk} \geqslant 0 \quad \text{for } 0 \leqslant i \leqslant n,\ \ 1 \leqslant j \leqslant n+1,\ \ 1 \leqslant k \leqslant m \tag{11}$$

$$\delta_{ik} \geqslant 0 \quad \text{for } 0 \leqslant i \leqslant n+1,\ \ 1 \leqslant k \leqslant m \tag{12}$$

The objective function (1) shows that this model minimizes routing cost, which consists of transportation fixed cost and variable cost. Constraints (2) and (3) form the feasible routes of vehicles, so that every customer is visited by exactly one vehicle (2), every vehicle that arrives to a customer must leave that customer (3), and vehicle is used to serve at most one route.

Constraints (5) and (6) explain the relationship between time variables and parameters in this model. Constraint (5) relates the starting service time of one customer with other customer. If vehicle k serving customer j after serving customer i ($x_{ijk}=1$), starting service time in customer j must be greater or equal to the sum of starting service time in customer i, the service time and transportation time from customer i to customer j ($\delta_{ik} + s_i + t_{ij} \leqslant \delta_{jk}$). Otherwise, there is no strict relationship between those starting service time (δ_{ik} and δ_{jk}) when $x_{ijk}=0$. Furthermore, δ_{0k} represents the time when vehicle k depart from the depot (it is assumed that a vehicle is ready to go at the beginning of a planning horizon, $s_0 = 0$) and $\delta_{n+1,k}$ represents the time when vehicle k return to the depot. Hence, the difference between the latter and the former represents the service/working duration of vehicle k and the limit of service duration is stated in constraint (6).

Vehicle load constraints are explained in (7)–(9). Constraint (7) 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 (8) 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 (9) balances the load of a vehicle after it serves a customer.

Constraints (10)–(12) state the domain of decision variables: all x_{ijk} are binary variables, y_{ijk} and δ_{ik} are non-negative real variables. Especially for δ_{ik}, it has the meaning of starting service time of customer i by vehicle k only when customer i are served by vehicle k ($x_{ijk} = 1$ and consequently $x_{jik} = 1$).

This formulation can be seen as a general model of VRPSPD. By setting the parameters, this model could lead to previously proposed model of VRPSPD. The formulation reduces to Min's [3] by setting the fixed cost $f = 0$, the variable cost $g = 1$, $d_{ij} = t_{ij}$ in Eq. (1), and the service duration limit of vehicle $D = \infty$. The formulation reduces to Dethloff's [4] by setting the fixed cost $f = 0$, variable cost $g = 1$, and service duration limit $D = \infty$. To reduce to Tang and Galvao's [7], set the fixed cost $f = 0$, the variable cost $g = 1$, the service time $s_i = 0$, define $t_{ij} = d_{ij}$, and define D as the maximum distance allowed per vehicle.

3. PSO for VRPSPD

In this section, a particle swarm optimization (PSO) algorithm is proposed for solving the general formulation of VRPSPD that is described in Section 2. Key features of the algorithm are explained in details including solution representation, decoding procedure to map representation to problem solution, and additional routine for searching appropriate number of vehicles.

3.1. PSO algorithm

As mentioned before, PSO is a population-based search method that imitated the physical movements of the individuals in the swarm as a searching method. In the PSO, a swarm of L particles is served as searching agent for a specific problem solution. A particle's position (Θ_l), which consists of H dimensions, is representing a solution of the problem. The ability of a particle to search for solution is represented by its velocity vector (Ω_l) which drives particle movement. In the PSO iteration step, every particle moves from one position to another position based on its velocity. Moving from one position to another, a particle is evaluating different prospective solutions of the problem.

PSO also imitated swarm's cognitive and social behavior as local and global search abilities. In the basic version of PSO, the particle's personal best position (Ψ_l) and the global best position (Ψ_g) are always updated and kept. The personal best position of a particle, which expresses the cognitive behavior, is defined as the position that gives the best objective function among the positions that have been visited by the particle. Once a particle reaches a position that has a better objective function than the previous best objective function for this particle (i.e. $Z(\Theta_l) < Z(\Psi_l)$), the personal best position is updated. The global best position, which expresses the social behavior, is the position that gives the best objective function among the positions that have been visited by all particles in the swarm. Once a particle reaches a position that has a better objective function than the previous best objective function for whole swarm (i.e. $Z(\Psi_l) < Z(\Psi_g)$), the global best position is also updated.

The personal best and global best positions are used for updating particle velocity. In each iteration step, the velocity Ω is updated based on three terms: inertia, cognitive learning and social learning terms. The inertia term forces a particle to move in the same direction as previous iteration. This term is calculated as a product of current velocity with an inertia weight (w). The cognitive term forces a particle to go back to its personal best position. This term is calculated as a product of a random number (u), personal best acceleration constant (c_p), and the difference between personal best position Ψ_l and current position Θ_l. The social term forces a particle to move to the global best position. This term is calculated as a product of a random number (u), global best acceleration constant (c_g), and the difference between global best position Ψ_g and current position Θ_l.

In the velocity-updating formula, random numbers are incorporated in order to randomize particle movement. Hence, two different particles may move to different position in the subsequent iteration even though they have similar position, personal best, and global best. Inertia weight and acceleration constants are the parameters that affect particle movement, each of them give the relative weight to the inertia, cognitive, and social term, respectively. High inertia weight means the particles tends to maintain current direction and low inertia weight means the particles tends to follow the cognitive and social term. It is common to have high inertia weight at the beginning of PSO iteration and low weight at the end, so that the particles are moving more freely to explore the solution space in the initial phase and following the cognitive and social term to exploit the personal best and global best in the final phase. It is expected that the particles can find a high quality personal and global best during the exploration phase, then the personal and global best can be used as good movement guidance in the exploitation phase.

In the PSO particle movement mechanism, it is also common to limit the search space of particle location, i.e. the position value of particle dimension is bounded at value $[\theta^{\min}, \theta^{\max}]$. This feature exists as the mechanism to avoid solution divergence. Hence, the position value of certain particle dimension is being set at the minimum or maximum value whenever it moves beyond the boundary. In addition, the velocity of corresponding dimension is reset to zero to avoid further movement beyond the boundary.

PSO works on finding the best position and the position is represented by a real number. To make PSO applicable to specific problem; the relationship between the position of particles and the solutions of that problem must be clearly defined. In VRP case, the particle's position represents the vehicle route. The details of the proposed solution representation and its relationship with vehicle route are described in the Sections 3.2 and 3.3.

A PSO algorithm for solving VRPSPD is proposed here based on the GLNPSO, a PSO Algorithm with multiple social learning structures [20]. In this PSO version, the component for social learning behavior includes not only the global best but also the local best (Ψ_l^L) and near neighbor best (Ψ_l^N). The local best is the best position of among several adjacent particles. The near neighbor best is a social learning behavior concept proposed by Veeramachaneni [21]. It is determined based on fitness-distance-ratio (FDR). The formula for determining these terms in the velocity updating formula is similar with the social term in the basic PSO, which is a product of a random number (u), an acceleration constant (c_l or c_n), and the difference between the social component (Ψ_l^L or Ψ_l^N) and current position Θ_l.

The details of the PSO algorithm for solving VRPSPD are presented below in Algorithm 1. In this algorithm, the particles are initialized in step 1, their corresponding fitness value are evaluated in steps 2–3, their cognitive and social information are updated in steps 4–7, and their positions are updated in step 8. Step 9 is the controlling step to repeat or stop the iteration. Note that the problem-specific steps are the conversion of particle's position into vehicle route in step 2, and, the determination of the performance measurement of the route in step 3.

Notation

τ	iteration index; $\tau = 1 \dots T$
l	particle index, $l = 1 \dots L$
h	dimension index, $h = 1 \dots H$
u	uniform random number in the interval $[0, 1]$
$w(\tau)$	inertia weight in the τth iteration
$\omega_{lh}(\tau)$	velocity of the lth particle at the hth dimension in the τth iteration
$\theta_{lh}(\tau)$	position of the lth particle at the hth dimension in the τth iteration
ψ_{lh}	personal best position (pbest) of the lth particle at the hth dimension
ψ_{gh}	global best position (gbest) at the hth dimension
ψ_{lh}^L	local best position (lbest) of the lth particle at the hth dimension
ψ_{lh}^N	near neighbor best position (nbest) of the lth particle at the hth dimension
c_p	personal best position acceleration constant
c_g	global best position acceleration constant
c_l	local best position acceleration constant
c_n	near neighbor best position acceleration constant
$\theta^{\max}$	maximum position value
$\theta^{\min}$	minimum position value
Θ_l	vector position of the lth particle, $[\theta_{l1}\ \theta_{l2}\ \cdots\ \theta_{lH}]$
Ω_l	vector velocity of the lth particle, $[\omega_{l1}\ \omega_{l2}\ \cdots\ \omega_{lH}]$
Ψ_l	vector personal best position of the lth particle, $[\psi_{l1}\ \psi_{l2}\ \cdots\ \psi_{lH}]$
Ψ_g	vector global best position, $[\psi_{g1}\ \psi_{g2}\ \cdots\ \psi_{gH}]$
Ψ_l^L	vector local best position of the lth particle, $[\psi_{l1}^L\ \psi_{l2}^L\ \cdots\ \psi_{lD}^L]$
R_l	the lth set of vehicle route
$Z(\Theta_l)$	fitness value of Θ_l
FDR	fitness-distance-ratio

Algorithm 1 (PSO Algorithm for VRPSPD).

1. Initialize L particles as a swarm, generate the lth particle with random position Θ_l in the range $[\theta^{\min}, \theta^{\max}]$, velocity $\Omega_l = 0$ and personal best $\Psi_l = \Theta_l$ for $l = 1 \dots L$. Set iteration $\tau = 1$.
2. For $l = 1 \dots L$, decode $\Theta_l(\tau)$ to a set of vehicle route R_l.
3. For $l = 1 \dots L$, compute the performance measurement of R_l, and set this as the fitness value of Θ_l, represented by $Z(\Theta_l)$.
4. Update pbest: For $l = 1 \dots L$, update $\Psi_l = \Theta_l$, if $Z(\Theta_l) < Z(\Psi_l)$.
5. Update gbest: For $l = 1 \dots L$, update $\Psi_g = \Psi_l$, if $Z(\Psi_l) < Z(\Psi_g)$.
6. Update lbest: For $l = 1 \dots L$, among all pbest from K neighbors of the lth particle, set the personal best which obtains the least fitness value to be Ψ_l^L.
7. Generate nbest: For $l = 1 \dots L$, and $h = 1 \dots H$, set $\psi_{lh}^N = \psi_{oh}$ that maximizing fitness-distance-ratio (FDR) for $o = 1 \dots H$. Where FDR is defined as

$$FDR = \frac{Z(\Theta_l) - Z(\Psi_o)}{|\theta_{lh} - \psi_{oh}|} \quad \text{where} \quad l \neq o \tag{13}$$

8. Update the velocity and the position of each lth particle:

$$w(\tau) = w(T) + \frac{\tau - T}{1 - T}[w(1) - w(T)] \tag{14}$$

$$\begin{aligned}\omega_{lh}(\tau+1) = w(\tau)\omega_{lh}(\tau) + c_p u(\psi_{lh} - \theta_{lh}(\tau)) + c_g u(\psi_{gh} - \theta_{lh}(\tau))\\ + c_l u(\psi_{lh}^L - \theta_{lh}(\tau)) + c_n u(\psi_{lh}^N - \theta_{lh}(\tau))\end{aligned} \tag{15}$$

$$\theta_{lh}(\tau+1) = \theta_{lh}(\tau) + \omega_{lh}(\tau+1) \tag{16}$$

If $\theta_{lh}(\tau+1) > \theta^{\max}$, then

$$\theta_{lh}(\tau+1) = \theta^{\max} \tag{17}$$

$$\omega_{lh}(\tau+1) = 0 \tag{18}$$

If $\theta_{lh}(\tau+1) < \theta^{\min}$, then

$$\theta_{lh}(\tau+1) = \theta^{\min} \tag{19}$$

$$\omega_{lh}(\tau+1) = 0 \tag{20}$$

9. If the stopping criterion is met, i.e. $\tau = T$, stop. Otherwise, $\tau = \tau + 1$ and return to step 2.

3.2. Solution representation

Solution representation of vehicle routes is one of the key elements for effective implementation of PSO for VRPSPD. An indirect representation is proposed here. It consists of two parts: the first part is related to the customers and the second part is related to the vehicles. This representation is decoded into vehicle routes by steps described in the next section.

The first part of the representation is required to set priority for customer to enter existing route in the route construction step. A random key with n elements is applied here. The first part of the representation consists of n dimensions of particle with each dimension assigned to a customer. The smaller value of the dimension corresponds to the higher priority to the customer.

The second part of the representation is based on the idea of vehicle route orientation. Route orientation of a vehicle is defined as a point in the service map that represents a certain area in which the vehicle is most likely to serve. Consequently, a vehicle route will tend to aggregate around its corresponding route orientation. A simple illustration of relationship between vehicle route and route

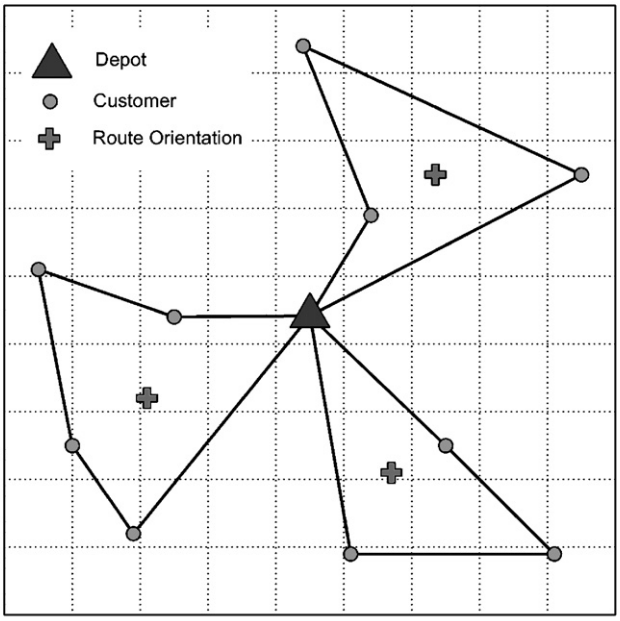

Fig. 1. Vehicle routes and route orientation.

orientation is depicted in Fig. 1. It is clearly seen that each vehicle covers certain service area that can be represented by the route orientation point.

A route orientation point is identified by its *x*–*y* coordinate in the service map. Since PSO uses position of particle with many dimensions to represent solution and the position values are a real numbers, each route orientation can be represented by two dimensions of a particle, one dimension for *x*-coordinate value and the other dimension for *y*-coordinate value. Hence, the second part of the representation would consist of 2*m* dimensions of particle that correspond to *m* available vehicles.

The route orientation is used as another basis for route construction. After all the route orientations are identified, preference of vehicles to serve each customer could be determined based on the distance of customer to orientation point. These preferences are set to ensure the spatial closeness among customers in one route, since the spatial closeness between customer and route orientation are maintained. While the spatial closeness is sustained, the total route distance might be shorter and the corresponding variable cost could be minimized.

The boundary of position mentioned in Algorithm 1, $[\theta^{\min}, \theta^{\max}]$, are determined based on the coverage of the service map. This boundary is very crucial for the particle's dimensions related to the vehicles, where the vehicle route orientation represented by these dimensions can be placed at every location in the service map. Hence, the minimum value of *x*-axis and *y*-axis of the service map is set as the minimum boundary $\theta^{\min}$ and the maximum value of *x*-axis and *y*-axis of the service map is set as the maximum boundary $\theta^{\max}$. This boundary has no effect for the particle's dimensions related to the customers; however, the same boundary values are also selected for these dimensions.

In summary, the proposed solution representation of VRPSPD with *n* customers and *m* vehicles will require particle with $(n+2m)$ dimension. Each particle dimension is encoded as a real number. The first *n* dimensions represent priorities of customers, each customer is represented by one dimension. The values in these dimensions are converted to customer priority list in the decoding step. The other 2*m* dimensions are related to vehicles, each vehicle is represented by two dimensions. These dimensions are extracted as the orientation point of vehicles in the Cartesian diagram/map. The summary of solution representation and its main conversion are displayed in Fig. 2.

3.3. Decoding method

Three steps must be taken in order to decode the proposed solution representation described in previous section into the VRPSPD solution. First, extract the information from the first *n* dimension to make a priority list of customers. Second, take the information from the last 2*m* dimension to determine the route orientation point of vehicles and use this information to create priority matrix of vehicles. Third, construct the vehicle routes based on the customer priority list and vehicle priority matrix.

In the first step, after the first *n* dimension of position value is removed, the customer priority list is constructed following the rule mentioned in previous sub-section. The simplest implementation of this rule is by sorting in ascending order the position value and taking the dimension index as the list.

The next step is to extract the route orientation point of vehicles and construct priority matrix of vehicle. The matrix is constructed based on the relative distance between these points and customers location. The distances can be calculated in every case of VRPSPD, since the location of customers is placed in a two-dimensional/Cartesian map. A customer is served first by vehicle with closer distance. Each row in the matrix keeps the vehicle priority for customers with the same priority.

The last decoding step is to construct a route based on the customer priority list and the vehicle priority matrix. One by one, each customer in the customer priority list is assigned to a vehicle based on its priority and such problem constraints as vehicle capacity constraint and service duration constraint. This newly assigned customer is inserted to the best position in the existing vehicle route based on the least additional cost. This is called the cheapest insertion heuristic. Another effort to improve solution quality of the route is to re-optimize the emerging route using some improvement heuristic methods such as 2-opt method. The details of this decoding procedure are described in Algorithm 2.

Algorithm 2 (Decoding method).

```
Decoding particle position ($\theta_{lh}$–position of the lth particle at
the hth dimension) into vehicle route ($R_{lj}$–route of the jth
vehicle corresponding to the lth particle)
1. Construct the priority list of customers ($U$)
  a. Build set $S = \{1, 2, \ldots, n\}$ and $U = \emptyset$
  b. Select $c$ from set $S$ where $\theta_{lc} = \min_{h\in S} \theta_{lh}$
  c. Add $c$ to the last position in set $U$
  d. Remove $c$ from set $S$
  e. Repeat step 1.b until $S = \emptyset$
2. Construct the vehicle priority matrix ($W$)
  a. Set the vehicle reference position. For $j = 1 \ldots m$,
     set $xref_j = \theta_{l,n+2j-1}$ and $yref_j = \theta_{l,n+2j}$
  b. For each customer $i$, $i = 1 \ldots n$
    i. Calculate the Euclidean distance between customer $i$ and vehicle route orientation points using following formula

       $$\Delta_j = \sqrt{(xpos_i - xref_j)^2 + (ypos_i - yref_j)^2} \qquad (21)$$

    ii.Build set $S = \{1, 2, \ldots, m\}$ and $W_i = \emptyset$
    iii. Select $c$ from set $S$ where $\Delta_c = \min_{j\in S} \Delta_j$
    iv. Add $c$ to the last position in set $W_i$
    v. Remove $c$ from set $S$
    vi. Repeat step 2.b.iii until $S = \emptyset$
```

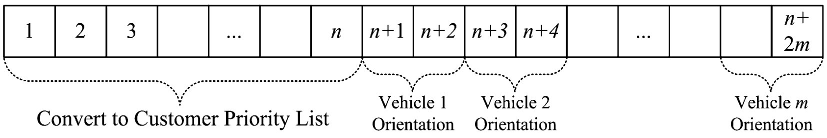

Fig. 2. Solution representation and its conversion.

```
3. Construct vehicle route
  a. Set $k = 1$
  b. Add customer one by one to the route
    i. Set $c = U_k$ and $b = 1$
    ii. Set $j = W_{c,b}$
    iii. Make a candidate of new route by inserting customer
         $c$ to the best sequence in the route $R_{lj}$ (route of vehicle$j$),
         which has the smallest additional cost (the cheapest
         insertion heuristic)
    iv. Check feasibility of the candidate route by evaluating
        all constraints: vehicle capacity and service duration
        constraints
    v. If a feasible solution is reached, update the route
       $R_{lj}$ with the candidate route and re-optimize emerging
       route with 2-opt method; then go to step 3.c
    vi. If $b = m$, go to step 3.c. Otherwise, set $b = b + 1$
        and repeat from step 3.b.ii
  c. If $k = n$, stop. Otherwise, set $k = k + 1$ and repeat step 3.b.
```

Following this decoding method, there is a possibility that a customer was not inserted into any routes. This situation is undesirable because some customers are not served and this corresponds to an infeasible solution. To avoid particle position that represents this kind of solution from being a candidate of best position, a large penalty is added to the fitness value of the particle for each customer not served. By this means and the principles of particle movement in the PSO, a particle that represents infeasible solution tends to move toward a position with lower degree of infeasibility and may eventually lead to a feasible solution.

3.4. *Searching for appropriate number of vehicles*

The proposed solution representation and decoding method is designed for solution of VRPSPD with fixed number of customers (n) and fixed number of vehicles (m). It is true that for most problems, the number of customers to be served is fixed and known in advance. However, the number of vehicles that actually serve the customers is a decision variable, which may have a value less than the number of available vehicles.

One advantage of this representation is its tendency to spread evenly the service area of vehicles. In other words, all available vehicles are more likely to be used to serve the customers. Since the variable fixed cost is the main contributor of the total cost, it is necessary to reduce the number of vehicles that are active in serving the customers. Hence, an additional routine is proposed here to obtain appropriate number of vehicle. This routine is implemented in the initialization step (Step 1) of PSO algorithm.

The main idea of this routine is to reduce the number of vehicles one by one while initializing a particle. Starting with the number of available vehicles, a solution representation is generated. After it is decoded, the number of customers served by each route is evaluated by trying to remove two particle dimensions corresponded to vehicle with smallest number of customers served. If the new representation leads to better fitness value, repeat the removal procedure until further removal lead to an inferior fitness value. Finally, the number of vehicles (m) is set to the number that gives the best fitness value. To speed up the process, this number is immediately used as the number of available vehicles in the process to generate subsequent particles. The detail of this routine is explained in Algorithm 3.

Algorithm 3 (Routine to search for appropriate number of vehicles).

1. Generate a random particle to represent n customers and m vehicles, which consists of $n + 2m$ dimensions. Set $\upsilon = m$.
2. Decode the particle into vehicle routes using Algorithm 2. Compute the performance measurement of the route, and set this value as the fitness value of the particle, Z.
3. Calculate the number of customers served by each vehicle.
4. Remove the corresponding dimensions of the vehicle with smallest number of customers. Reduce the particle size by two dimensions. Set $\upsilon = \upsilon - 1$.
5. Decode the updated particle into vehicle routes using Algorithm 2. Compute the performance measurement of the route, and set this value as the fitness value of the updated particle, Z'.
6. If the fitness value of updated particle is smaller than its of original particle, $Z' < Z$, set $Z = Z'$, then repeat step 3–5; otherwise, go to step 7.
7. Add back the two particle dimensions last removed. Increase the particle size by two dimensions. Set $\upsilon = \upsilon + 1$. Then, set the new value of number of available vehicles $m = \upsilon$ for subsequent particles.

4. Computational result

4.1. *Comparison with results from literature*

Computational experiments are conducted by applying this proposed algorithm to some benchmark data sets of VRPSPD in order to evaluate the performance of the proposed method. The first benchmark data set is the data introduced by Dell'Amico et al. [9], where the problem size is less than 40 customers. By using this data, the PSO algorithm can be compared against the exact solution method of Dell'Amico et al. [9]. The second benchmark data set is the data introduced by Dethloff [4], which comprises four sets 50-customer problems. The PSO performance can be evaluated across some heuristic that had been tested on these problems, including heuristic of Dethloff [4], Tang and Galvao [7] and Bianchessi and Righini [8]. The last benchmark data set is the data introduced by Nagy and Salhi [6], which consists of five sets of problems with 50–199 customers. Using this data set, the PSO results can be compared with those results from Dethloff [4], Nagy and Salhi [6], and Tang and Galvao [7].

The algorithm is implemented in C# language using Microsoft Visual Studio.NET 1.1 on a PC with Intel P4 3.4 GHz—1 GB RAM. For each data set, 10 replications of the algorithm are tried. The PSO parameters are set based on the result of some preliminary experiments that are carried out to observe the behavior of algorithm in different parameter setting. The PSO parameters are summarized in Table 1.

4.1.1. *Dell'Amico data*

The first computational experiment is conducted on the benchmark data set of Dell'Amico et al. [9] which comprises five classes

Table 1
Summary of PSO parameters.

Parameter	Value
Number of particle	$L = 50$
Number of iteration	$T = 1000$
Number of neighbor	$K = 5$
First inertia weight	$w(1) = 0.9$
Last inertia weight	$w(T) = 0.4$
Personal best position acceleration constant	$c_p = 1$
Global best position acceleration constant	$c_g = 0$
Local best position acceleration constant	$c_l = 1$
Near neighbor best position acceleration constant	$c_n = 2$

Table 2
Comparison of Dell'Amico et al. and PSO solution.

Instance class	Average total cost	
	Dell'Amico et al. [9]	PSO
Class 1	522.5	524.7
Class 2S	236822.7	236826.4
Class 2C	341481.2	341994.3
Class 3S	12082.3	11912.3
Class 3C	15979.6	15984.6

of VRPSPD instances. Each class consists of instances with 20 and 40 customers. Class 1 consists of 12 instances, while Class 2S, 2C, 3S, and 3C consist of 18 instances. The VRPSPD of these instances is similar to the Dethloff formulation. Hence, the following problem parameters are required to be set in the proposed PSO method: fixed cost per vehicle, $f = 0$; variable cost per distance unit, $g = 1$; service duration limit $D = \infty$; and the number of available vehicles is equal to the number of available vehicle in the optimal/best known solution.

The comparison of the best solution among 10 PSO iterations with the upper bound result of Dell'Amico et al. [9] is presented in Table 2, in which the average total cost of the instances in each class is compared. It is noted that Dell'Amico et al. [9] already provided the optimal solution of 75 out of 84 instances. Hence, the upper bound represents the optimal solution or the best solution found. The result presented in Table 2 implies that the proposed PSO method is able to provide high quality solutions that are very close to the optimal solution. Moreover, these solutions can be obtained in very short computational time, in average of 9 and 27 s, respectively, for 20-customer and 40-customer problems.

4.1.2. Dethloff data

The second computational experiment is conducted on the benchmark data set of Dethloff [4] which comprises four data sets named SCA3, SCA8, CON3, and CON8. Each data set consists of 10 instances of a 50-customer problem with specific characteristics: SCA data sets are generated with customers scattered uniformly in the service region, CON data sets are generated with half of the customers located uniformly in the service region and the other half are concentrated in certain part of the service region. The number after SCA or CON indicated the parameter for determining vehicle capacity.

As mentioned earlier, the VRPSPD is formulated by Dethloff [4] as the problem to minimize the total traveled distance subject to maximum capacity constraint of the vehicle. Hence, the following problem parameters are set as follows: fixed cost per vehicle, $f = 0$; variable cost per distance unit, g=1; service duration limit D=∞; and the number of available vehicles is equal to the number of available vehicle in the best known solution.

Table 3
Comparison of several methods on Dethloff data.

Set	Average total cost			
	Dethloff [4]	Tang and Galvao [7]	Bianchessi and Righini [8]	PSO
SCA3	746.6	674.2	684.6	675.8[a,c]
SCA8	1166.4	1044.4	1035.7	1041.8[a,b]
CON3	597.3	564.2	568.5	569.6[a]
CON8	860.6	774.3	776.4	798.3[a]

[a] PSO result better than Dethloff result.

[b] PSO result better than Tang and Galvao result.

[c] PSO result better than Bianchessi and Righini result.

The comparison of the best solution among 10 PSO iterations with the result from Dethloff [4], Tang and Galvao [7] and Bianchessi and Righini [8] is presented in Table 3. To make a direct comparison across these existing results, only the average result over the 10 instances of each data set is reported. It is shown that the proposed PSO method is competitive with existing methods: the PSO result outperforms the Dethloff result for all data set, better than the Tang and Galvao result for SCA8 data set, and better than the Bianchessi and Righini result for SCA3 data set. More over, only small difference between PSO and the best result is observed for other cases. In addition to this result, the proposed PSO gives a reasonable computational time for solving this benchmark data in which approximately only 30 s of computational time is required for each instance.

4.1.3. Nagy and Salhi data

Another computational experiment is conducted by applying this proposed algorithm to the benchmark data set of Nagy and Salhi [6] which modified the basic VRP benchmark data set of Christofides [1] to be the benchmark data set of VRPSPD. The corresponding depot and customer coordinate remained the same, but the original delivery demand data on basic VRP benchmark is split into pickup quantity and delivery quantity. There are five new sets of data based on the splitting method of demand data, and the new problem sets are named T, Q, H, X, and Y.

Since the proposed method is implemented for the general formulation of VRPSPD, problem parameters need to be set in order to compare with results from previous works of VRPSPD. The following problem parameters are used: fixed cost per vehicle, $f = 0$; variable cost per distance unit, $g = 1$; traveling time is equal to corresponding traveled distance, $t_{ij} = d_{ij}$; and the number of available vehicles is equal to the number of available vehicle in the best known solution. The result of problem T, Q, and H are compared with the best results from Salhi and Nagy [5], which is the only result found in the literature. The result of problem X and Y are compared with the best solution among Nagy and Salhi [6], Dethloff [4], Tang and Galvao [7]. However, result of Tang and Galvao [7] for instances number 6–10 and 13–14 are omitted from comparison because the different problem setting in which the customer service time are not considered.

In order to compare the solution obtained with the best-known solution, percentage of deviation from best-known solution is used. The formula for calculating the percentage of deviation is as follows:

$$\%dev = \frac{Z - Z*}{Z*} \times 100\% \tag{22}$$

where $\%dev$: percentage of deviation from best-known solution; Z: objective function of current solution; $Z*$: objective function of best known solution.

The comparison of the best solution among 10 PSO iterations and the best-known solutions are shown in Tables 4 and 5. This comparison shows that the results from the proposed PSO algorithm are competitive with other published results. As shown in Table 4, in almost all instances of the problem T, Q, and H, the best objective function of PSO results are better than the corresponding best-known solutions. Additionally from Table 5, the PSO result of sixteen instances of problem X and Y are better than its corresponding best-known solution. The detail of these new best solutions of VRPSPD instances are presented in the following webpage:

Table 4
Comparison of best-known solution and best PSO solution of T, Q and H Instances with fixed cost $f = 0$ and variable cost $g = 1$.

Instance	Best-known solution [5]		Best PSO solution		
	No. vehicle	Total cost (distance)	No. vehicle	Total cost (distance)	% dev
CMT1T	5	541	5	**520**	−3.9
CMT2T	10	839	9	**810**	−3.5
CMT3T	10	903	7	**827**	−8.5
CMT4T	13	1111	11	**1014**	−8.8
CMT5T	18	1423	15	**1297**	−8.9
CMT6T	6	571	6	**555**	−2.7
CMT7T	–	–	12	**942**	–
CMT8T	10	911	9	**904**	−0.7
CMT9T	14	1164	14	1206	3.6
CMT10T	18	1418	18	1501	5.8
CMT11T	7	1075	7	**1026**	−4.5
CMT12T	10	827	9	**792**	−4.3
CMT13T	12	1600	11	**1548**	−3.3
CMT14T	11	866	10	**846**	−2.3
CMT1Q	5	557	4	**490**	−12.1
CMT2Q	11	860	8	**739**	−14.1
CMT3Q	9	918	6	**768**	−16.4
CMT4Q	14	1164	9	**938**	−19.4
CMT5Q	19	1477	13	**1174**	−20.5
CMT6Q	6	594	6	**557**	−6.3
CMT7Q	–	–	12	**933**	–
CMT8Q	9	918	9	**890**	−3.0
CMT9Q	15	1178	14	1214	3.1
CMT10Q	19	1477	18	1509	2.2
CMT11Q	7	1075	6	**964**	−10.3
CMT12Q	10	843	7	**733**	−13.1
CMT13Q	13	1613	11	**1570**	−2.7
CMT14Q	11	873	10	**825**	−5.5
CMT1H	6	594	3	**464**	−21.8
CMT2H	12	873	6	**668**	−23.4
CMT3H	9	915	4	**701**	−23.4
CMT4H	14	1164	6	**883**	−24.1
CMT5H	19	1509	9	**1044**	−30.8
CMT6H	6	594	6	**557**	−6.3
CMT7H	–	–	11	**943**	–
CMT8H	9	915	9	**899**	−1.7
CMT9H	14	1164	14	1207	3.7
CMT10H	19	1509	19	**1499**	−0.7
CMT11H	8	1120	4	**830**	−25.9
CMT12H	11	850	5	**635**	−25.3
CMT13H	11	1546	11	1565	1.2
CMT14H	11	866	10	**824**	−4.8

Table 5
Comparison of best-known solution and best PSO solution of X and Y instances with fixed cost $f = 0$ and variable cost $g = 1$.

Instance	Best-known solution			Best PSO solution		
	No. vehicle	Total cost (distance)	Ref.	No. vehicle	Total cost (distance)	% dev
CMT1X	3	472	[7]	3	**467**	−1.1
CMT1Y	3	470	[7]	3	**467**	−0.7
CMT2X	7	695	[7]	6	710	2.1
CMT2Y	7	700	[7]	6	710	1.5
CMT3X	5	721	[7]	5	738	2.3
CMT3Y	5	719	[7]	5	740	3.0
CMT4X	7	880	[7]	7	912	3.7
CMT4Y	7	878	[7]	7	913	4.0
CMT5X	11	1098	[7]	10	1167	6.3
CMT5Y	10	1083	[7]	10	1142	5.5
CMT6X	6	584	[4]	6	**557**	−4.7
CMT6Y	6	584	[4]	6	**557**	−4.7
CMT7X	11	961	[4]	11	**919**	−4.3
CMT7Y	11	961	[4]	11	**934**	−2.8
CMT8X	10	923	[5]	9	**896**	−2.9
CMT8Y	10	923	[5]	9	**902**	−2.3
CMT9X	15	1215	[5]	15	1225	0.8
CMT9Y	15	1215	[5]	15	1230	1.3
CMT10X	19	1571	[4]	19	**1520**	−3.3
CMT10Y	20	1527	[4]	18	**1485**	−2.8
CMT11X	4	900	[7]	4	**895**	−0.5
CMT11Y	5	910	[7]	4	**900**	−1.1
CMT12X	6	675	[7]	5	691	2.4
CMT12Y	6	689	[7]	5	697	1.2
CMT13X	11	1576	[4]	11	**1560**	−1.0
CMT13Y	11	1576	[4]	11	**1568**	−0.5
CMT14X	10	871	[4]	10	**826**	−5.2
CMT14Y	10	871	[4]	10	**823**	−5.5

Table 6
Statistical summary of PSO result on X and Y instances.

Instance	No. cust.	Average total cost	Standard deviation	% Standard deviation	Average comp. time (second)
CMT1X	50	469.57	2.95	0.6	40
CMT1Y	50	468.88	2.77	0.6	40
CMT2X	75	717.45	6.85	1.0	54
CMT2Y	75	716.70	5.13	0.7	54
CMT3X	100	746.20	4.08	0.5	114
CMT3Y	100	746.78	3.76	0.5	113
CMT4X	150	928.20	7.98	0.9	207
CMT4Y	150	926.12	6.75	0.7	204
CMT5X	199	1196.13	17.39	1.5	285
CMT5Y	199	1182.67	22.74	1.9	286
CMT6X	50	558.53	1.02	0.2	28
CMT6Y	50	559.12	1.03	0.2	28
CMT7X	75	944.13	17.80	1.9	66
CMT7Y	75	956.90	15.01	1.6	65
CMT8X	100	912.87	10.69	1.2	99
CMT8Y	100	918.56	10.27	1.1	100
CMT9X	150	1246.24	10.95	0.9	189
CMT9Y	150	1245.14	14.31	1.1	187
CMT10X	199	1529.45	6.28	0.4	323
CMT10Y	199	1516.23	22.09	1.5	323
CMT11X	120	915.30	15.69	1.7	226
CMT11Y	120	913.77	10.53	1.2	228
CMT12X	100	716.80	24.75	3.5	115
CMT12Y	100	728.74	25.18	3.5	114
CMT13X	120	1575.61	10.48	0.7	135
CMT13Y	120	1578.18	7.89	0.5	135
CMT14X	100	830.04	4.84	0.6	98
CMT14Y	100	829.93	5.25	0.6	97

http://ind.uajy.ac.id/~jinai/PSO_for_VRPSPD_COR_Appendix.htm. Furthermore, for all instances that are worse than the best-known solution, the biggest deviation is about six percent.

Statistics of the PSO result on problems X and Y are presented in Table 6. It comprises the average and standard deviation of the objective function, percentage of standard deviation over the average, and the average computational time. The robustness of the proposed method in term of solution quality is implied in this statistics. Even though the proposed method is a random search algorithm, the variation of solutions over replications are very consistent as demonstrated by the small standard deviation.

The computational results show that the computational time of the proposed method tends to be linearly proportional with the number of customers, which is representing the problem size. This relation is desirable, since it is only require linear additional time to apply this method on a bigger size problem.

The high-quality result yielded by the proposed method might come from two factors: the idea of vehicle orientation and the heuristic for constructing routes. The implementation of vehicle orientation will ensure the spatial closeness of customers that are included in the route. Hence, the constructed route will cover only a relatively narrow area. The customer is inserted into the best position in an existing route by applying the route construction heuristics. Furthermore, the 2-opt method is capable of improving a newly constructed route. The combinations of these efforts are potential for yielding a good solution.

The simplicity of PSO may also contribute to the performance. By its mechanism, the particles are able to explore various areas in the searching space within a few computational steps. It means that diverse solutions of vehicle routes are generated during the iteration process, since one particle corresponds to one solution of vehicle routes. This diversification of solutions will increase the possibility

Table 7
Comparison of best-known solution and best PSO solution of X and Y instances with fixed cost $f=100$ and variable cost $g=1$.

Instance	Best-known solution				Best PSO solution			
	No. vehicle	Total distance	Total cost	Ref.	No. vehicle	Total distance	Total cost	% dev
CMT1X	3	472	772	[7]	3	467	**767**	−0.7
CMT1Y	3	470	770	[7]	3	467	**767**	−0.4
CMT2X	7	695	1395	[7]	6	707	**1307**	−6.3
CMT2Y	7	700	1400	[7]	6	709	**1309**	−6.5
CMT3X	5	721	1221	[7]	5	742	1242	1.7
CMT3Y	5	719	1219	[7]	5	739	1239	1.7
CMT4X	7	880	1580	[7]	7	923	1623	2.7
CMT4Y	7	878	1578	[7]	7	920	1620	2.7
CMT5X	11	1098	2198	[7]	10	1150	**2150**	−2.2
CMT5Y	10	1083	2083	[7]	10	1138	2138	2.7
CMT6X	6	584	1184	[4]	6	557	**1157**	−2.3
CMT6Y	6	584	1184	[4]	6	557	**1157**	−2.3
CMT7X	11	961	2061	[4]	11	931	**2031**	−1.4
CMT7Y	11	961	2061	[4]	11	933	**2033**	−1.4
CMT8X	9	928	1828	[4]	9	902	**1802**	−1.4
CMT8Y	9	936	1836	[4]	9	906	**1806**	−1.6
CMT9X	15	1215	2715	[5]	15	1229	2729	0.5
CMT9Y	15	1215	2715	[5]	15	1237	2737	0.8
CMT10X	19	1571	3471	[4]	19	1499	**3399**	−2.1
CMT10Y	19	1571	3471	[4]	19	1485	**3385**	−2.5
CMT11X	4	900	1300	[7]	4	898	**1298**	−0.2
CMT11Y	5	910	1410	[7]	4	904	**1304**	−7.5
CMT12X	6	675	1275	[7]	5	682	**1182**	−7.3
CMT12Y	6	689	1289	[7]	5	681	**1181**	−8.4
CMT13X	11	1576	2676	[4]	11	1570	**2670**	−0.2
CMT13Y	11	1576	2676	[4]	11	1568	**2668**	−0.3
CMT14X	10	871	1871	[4]	10	824	**1824**	−2.5
CMT14Y	10	871	1871	[4]	10	824	**1824**	−2.5

to find a high-quality solution. In addition, the searching mechanism of the particle also fits for searching the vehicle orientations, in which the best point of a vehicle orientation is explored around current best points. Furthermore, PSO always keeps and uses the information on the best position of the particles to direct the particles movement. Consequently, the iteration process of PSO may end with a high-quality solution within a relatively short computational time.

4.2. Effect of fixed cost

The computational results in Section 4.1 are inline with the previous work in VRPSPD, in which ignoring the fixed cost of vehicle. However, literature on other areas of VRP have acknowledged the importance of the fixed cost of vehicle and considered it as one part of the objective function [1]. The proposed method can straightforwardly handle this situation, since the mathematical model considered the fixed cost in the objective function.

Another computational experiment is carried out in order to show the performance of the proposed method when considering fixed cost. All PSO and problem parameters are the same as those used in the experiment in Section 4.1.3, except that the fixed cost is set as 100. The result of this experiment for problems X and Y are shown in Table 7. For comparison purpose, the total cost of the best result of previous works [4,5,7] are recalculated using similar unit cost of fixed cost $f=100$ and variable cost $g=1$ before to be selected as the best-known solution.

It is shown in Table 7 that the computational PSO result for 21 instances of problems X and Y are better than the best-known solution. By using this amount of fixed cost, it implies that the smaller number of vehicles leads to the better total cost. Hence, it is not a surprise that the proposed method result is frequently better than the best-known solution since a special effort for reducing the number of vehicles (Algorithm 3) is included. This effort will ensure that the number of vehicles is as small as possible. As a result, the number of vehicles from the PSO result is at most exactly the same as the best-known solution and in some cases smaller than those of the best-known solutions.

5. Conclusion and further study

A generalized formulation of VRPSPD for three existing formulations in the literature is presented in this paper along with a solution method based on PSO algorithm. The computational result shows that the proposed PSO method is effective for solving the VRPSPD. The effectiveness of the method comes from the combination of following reasons. First, the idea of vehicle orientation makes each of routes only cover a restricted area. Second, the solution quality is improved from the cheapest insertion heuristic and 2-opt method which are applied during the route construction. Third, a special algorithm reduces the number of vehicles that actually serve the customers. Fourth, the mechanism of PSO that can generate diverse solutions and keep the best solution found during the iteration process.

Some aspects may further improve the performance of the proposed algorithm, such as parameter optimization and programming implementation. Although the PSO parameter set used in this paper came from some preliminary experiment, it may not be the best one. In addition, the programming implementation of the algorithm may be further optimized. Since these efforts may yet contribute to additional performance gains in both the solution quality and computational time, a further study on these aspects is still necessary.

Some further research to apply the proposed method to other VRP variants should be carried out to show generality of the method. Since the variants of VRP differ from one another only on the specific problem constraints, the adjustment is only required in the constraint feasibility checking of the decoding method. However, the effectiveness of this idea needs further exploration.

Acknowledgements

The research is part of the research program of the High Performance Computing Group at Asian Institute of Technology (AITHPC) supported by the Thai GRID project. The authors wish to thank AITHPC and the Thai GRID Center for the access of computing facility and the technical support.

The authors express thanks to Prof. Dr. Jan Dethloff, Prof. Dr. Mauro Dell'Amico, and Dr. Matteo Salani for providing the data sets used in this paper, also to the anonymous reviewers for the valuable comments and suggestions on this paper preparation.

The first author expresses gratitude to Universitas Atma Jaya Yogyakarta, Indonesia for providing the financial support for his study in Asian Institute of Technology, Thailand.

References

[1] Christofides N, Mingozzi A, Toth P. The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. Combinatorial optimization. Chicester: Wiley; 1979. p. 315–38.

[2] In: Toth P, Vigo D, editors. The vehicle routing problem. Philadelphia, PA: SIAM; 2002.

[3] Min H. The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research A 1989;23(5):377–86.

[4] Dethloff J. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. OR Spektrum 2001;23(1):79–96.

[5] Salhi S, Nagy G. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the Operational Research Society 1999;50(10):1034–42.

[6] Nagy G, Salhi S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research 2005;162(1):126–41.

[7] Tang FA, Galvao RD. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Computers and Operations Research 2006;33(3):595–619.

[8] Bianchessi N, Righini G. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers and Operations Research 2007;34(2):578–94.

[9] Dell'Amico M, Righini G, Salani M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transportation Science 2006;40(2):235–47.

[10] Berbeglia G, Cordeau JF, Gribkovskaia I, Laporte G. Static pickup and delivery problems: a classification scheme and survey. TOP 2007;15(1):1–31.

[11] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of IEEE international conference on neural networks, vol. 4, 1995. p. 1942–8.

[12] Kennedy J, Eberhart RC. Swarm intelligence. San Francisco: Morgan Kaufmann Publishers; 2001.

[13] Clerc M. Particle swarm optimization. London: ISTE; 2006.

[14] Baker BM, Ayechew MA. A genetic algorithm for the vehicle routing problem. Computers and Operations Research 2003;30(5):787–800.

[15] Berger J, Barkaoui M. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Computers and Operations Research 2004;31(12):2037–53.

[16] Thangiah SR, Salhi S. Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem. Applied Artificial Intelligence 2001;15(4): 361–83.

[17] Bullnheimer R, Hartl F, Strauss C. An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research 1999;89: 319–28.

[18] Reimann M, Doerner K, Hartl RF. D-ants: savings based ants divide and conquer the vehicle routing problem. Computers and Operations Research 2004;31(4):563–91.

[19] Chen AL, Yang GK, Wu ZM. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University Science A 2006;7(4):607–14.

[20] Pongchairerks P, Kachitvichyanukul V. A non-homogenous particle swarm optimization with multiple social structures. In: Proceedings of international conference on simulation and modeling. 2005. p. A5-02.

[21] Veeramachaneni K, Peram T, Mohan C, Osadciw LA. Optimization using particle swarms with near neighbor interaction. In: Proceedings of genetic and evolutionary computation conference. 2003. p. 110–21.