Paper 03

by The Jin Ai

Submission date: 19-Jul-2019 02:38PM (UTC+0700)

Submission ID: 1153159534

File name: Paper_03_IJOR_A_PSO_for_VRP_with_Time_Windows.pdf (638.19K)

Word count: 8747

Character count: 40603

A Particle Swarm Optimisation for Vehicle Routing Problem with Time Windows

The Jin Ai*

Department of Industrial Engineering,
Universitas Atma Jaya Yogyakarta,
Jl. Babarsari 43, Yogyakarta 55281, Indonesia
E-mail: jinai@mail.uajy.ac.id
*Corresponding author

Voratas Kachitvichyanukul

School of Engineering and Technology,
Asian Institute of Technology,
P.O. Box 4, Klong Luang, Pathumtani 12120, Thailand
E-mail: voratas@ait.ac.th

Abstract: A heuristic based on Particle Swarm Optimisation (PSO) algorithm for solving VRPTW, which is an extension of PSO application for the Capacitated Vehicle Routing Problem (CVRP) (Ai and Kachitvichyanukul, 2007), is presented in this paper. A computational experiment is carried out by running the proposed algorithm with the VRPTW benchmark data set of Solomon (1987). The results show that the proposed algorithm is able to provide VRPTW solutions that are very close to its optimal solutions for problems with 25 and 50 customers within reasonably short of computational time

Keywords: PSO; particle swarm optimisation; VRPTW; vehicle routing problem with time windows; heuristics.

Reference to this paper should be made as follows: Ai, T.J., Kachitvichyanukul, V. (2009) 'A Particle Swarm Optimisation for Vehicle Routing Problem with Time Windows', *Int. J. Operational Research*, Vol. 6, No. 4, pp.519–537.

Biographical notes: The Jin Ai is a lecturer in Industrial Engineering, Faculty of Industrial Technology, Universitas Atma Jaya Yogyakarta, Indonesia. He received a DEng from the School of Engineering and Technology at Asian Institute of Technology, Thailand in 2008. His research interests include evolutionary computation and its application on industrial systems. Currently, he is working on the application of particle swarm optimisation on vehicle routing problems.

Voratas Kachitvichyanukul is an Associate Professor in Industrial Engineering and Management, School of Engineering and Technology, Asian Institute of Technology, Thailand. He received a PhD from the School of Industrial Engineering at Purdue University in 1982. He has extensive experiences in simulation modelling of manufacturing systems. He had worked for FORTUNE 500 Companies such as Compaq Computer Corporation and Motorola Incorporated. He had also worked for SEMATECH as technical coordinator of the future factory program. His teaching and research interests include planning and scheduling, high-performance computing and applied operations research with special emphasis on large-scale industrial systems.

1 Introduction

Nowadays, most of the companies are still using truck as their inland distribution mode because of its flexibility. In this era of increasing fuel price, the distribution function must be managed efficiently. Inefficiency in distribution function, i.e., inefficient vehicle routes means more energy consumption and higher distribution cost. Consequently, the problem to find an efficient vehicle route for distributing goods from depot to customers, which is usually called a Vehicle Routing Problem (VRP), is very critical in this field.

The main problem of this paper is a specific VRP, which is commonly found in distribution practice, the Vehicle Routing Problem with Time Windows (VRPTW). In this problem, customers may receive the delivery service from vehicle only in a certain time interval, the customer time windows. This additional requirement makes the problem more difficult to solve. In terms of computational complexity, this problem is well known to be an NP-hard problem. As a consequence, exact solution method can only find optimal solution in a reasonable amount of time for small size problems (Kohl and Madsen, 1997; Bard et al., 2002). Therefore, research in this area has been focusing on developing approximate solution methodology for the problem, i.e., heuristics.

In the literature, population-based methods such as Ant Colony Optimisation (ACO) (Gambardella et al., 1999) and Genetic Algorithm (GA) (Tan et al., 2001; Berger and Barkaoui, 2004; Alvarenga et al., 2007) were widely applied to solve VRPTW. Specifically for GA, the applications of this method to this problem become more and more complex, since this method had been explored for quite some time. It is true that the solution quality is improved by more complex method, but the computational effort is also increased.

As an emerging population-based method, Particle Swarm Optimisation (PSO) has not been well explored for solving VRPTW. Qing et al. (2006) tried to apply PSO for VRPTW and only showed the performance of this method on very small problem case consisting of eight customers, which is very far from realistic problem size. However, in broader area of vehicle routing, a simple continuous version of PSO has shown a promising result for solving the basic version of VRP, the Capacitated Vehicle Routing Problem or CVRP (Ai and Kachitvichyanukul, 2007). Therefore, the exploration of PSO capability for solving VRPTW is an open research gap in this area.

This paper fills this gap by presenting a PSO for solving VRPTW, which is an extension of recent work on PSO for CVRP. Since the VRPTW is different from the CVRP only by the time window constraints, the main components of PSO algorithm that have been successfully applied to the CVRP are also used in the proposed algorithm. The proposed method uses a simple continuous version of PSO to avoid high level of computational complexity and long computational time.

The remaining parts of this paper are organised as follows: Section 2 presents the literature review on VRPTW and PSO; Section 3 reviews PSO framework for solving

VRPTW; Section 4 explains the proposed solution representation and decoding methods; Section 5 discusses the computational experiment of the PSO on VRPTW benchmark data set; finally, Section 6 concludes the result of this research.

2 Literature review

2.1 *Vehicle Routing Problem with Time Windows*

The VRPTW is a variant of VRP, which attempts to determine a set of vehicle routes that involves a single depot, a fleet of identical vehicles stationed at the depot, and a set of customers who require delivery of goods from the depot. The objective of VRPTW is to minimise the total travelled time or distance, subject to maximum working time of vehicles, capacity of vehicles, and time window of customers. Time window of a customer is a time interval in which the customer is ready to be served. A complete survey on VRPTW formulation and solution techniques is provided by Bräysy and Gendreau (2005a, 2005b).

Exact optimisation and heuristic approaches have been applied to solve the VRPTW. Since the VRPTW is an NP-hard problem, the exact approach is effective for only small size problems (i.e., Kohl and Madsen, 1997; Bard et al., 2002). Hence, heuristics are favourable for practical applications that usually involve large number of customers. Many heuristic methods have been proposed for solving VRPTW, including the population-based search method such as GA and ACO. In recent applications listed below, the basic method had been enhanced to achieve better performance in solving the problem.

Gambardella et al. (1999) proposed an ACO-based approach to solve VRPTW. Their ACO consisted of two artificial ant colonies designed to optimise two different objective functions. These two colonies are cooperated through exchanging information towards a good solution in terms of both objectives. They showed that the proposed method was competitive with the best known existing methods, in terms of solution quality and computation time.

Tan et al. (2001) proposed a hybrid GA for solving VRPTW that combines the simple GA with local search methods. They applied the heuristic to solve Solomon's benchmark instances, and yield some solutions that improved the best published solution in literature.

Berger and Barkaoui (2004) presented parallel version of hybrid GA for the VRPTW. Their main algorithm consisted of two populations of solutions focusing on separate objectives, specially designed genetic operator based on some successful routing techniques, and parallel computing mechanism to speed up the operators. Their computational results showed that the proposed algorithm was very competitive with the existing procedure and provided some new best-known solutions.

Alvarenga et al. (2007) proposed a two-stage heuristic approach for solving the VRPTW. Their approach was involving an efficient GA and a set partitioning formulation. Their computational results outperformed the existing heuristic methods in terms of the minimal travel distance.

2.2 Particle Swarm Optimisation

PSO is a population-based search method proposed by Kennedy and Eberhart (1995). The main motivation came from the behaviour of group organisms such as bee swarm, fish school, and bird flock. PSO imitates the physical movements of the individuals in the swarm as well as its cognitive and social behaviour as a searching method. In PSO, a problem solution is represented by the position value of a multi-dimensional particle. Particle's velocity represents particle searching ability. The basic version of PSO algorithm starts by initialising the population of particles, which is usually called a swarm, with random position and velocity. In each iteration step, every particle moves to a new position following its velocity; and its velocity is updated based on its personal and global best position. Personal best position of a particle, which expresses the cognitive behaviour of a particle, is defined as the best position found by the particle. It will be updated whenever the particle reaches a position with better fitness value than the fitness value of the previous local best. Global best position, which expresses the social behaviour, is defined as the best position found by the whole swarm. It will be updated whenever a particle reaches a position with better fitness value than the fitness value of the previous global best. Comprehensive details of the PSO mechanism, technique, and applications are provided by Kennedy and Eberhart (2001) and also Clerc (2006).

As an emerging population-based search method, PSO has been recently applied to many operational research problems, such as flow shop scheduling, job shop scheduling, home care worker scheduling, multi-mode resource-constrained project scheduling, and vehicle routing. Some of the applications are briefly described below.

Tasgetiren et al. (2007) applied PSO to solve the flow shop scheduling problem. They used a heuristic called the Smallest Position Value (SPV) to transform position value of particle to a job sequence. They also incorporated a local search procedure called Variable Neighborhood Search (VNS) to further improve the performance of PSO. The proposed PSO was proven to improve some best-known solutions of benchmark problems with total flow time and makespan criterions. However, Pongchairerks and Kachitvichyanukul (2007) found that in the method of Tasgetiren et al. (2007), the main contributor to solution quality is the time-consuming VNS and not from PSO.

Sha and Hsu (2006) proposed a hybrid PSO for the job shop scheduling problem. They modified the PSO algorithm by applying preference list-based representation for particle position, swap operator for particle movement, and tabu list concept for particle velocity. They used Giffler and Thompson's heuristic to decode a particle position into a schedule and applied tabu search to improve the solution quality. They showed that the hybrid PSO is better than other traditional metaheuristics for solving the job shop scheduling problem.

Other authors also developed special structure of PSO systems that had been applied for solving job shop scheduling problem. The first structure is called two-level PSO, in which the PSO algorithm is being executed in two levels for solving the problem (Pongchairerks and Kachitvichyanukul, 2009), and the second structure is called two-stage PSO, in which the PSO algorithm is being executed in two stages for solving the multi-objective problem (Pratchayaborirk and Kachitvichyanukul, 2007). Both applications performed consistently well in terms of solution quality and computational time when it is evaluated using the standard benchmark problems.

Akjiratikarl et al. (2007) applied PSO to solve the home care worker scheduling problem. They used continuous PSO algorithm for this problem by defining particle as a multi-dimensional point in space, which represents the corresponding care activities and assignment priority. A special heuristic assignment scheme is proposed to transform a particle to the discrete job schedule. They also placed local improvement procedures, such as insertion and swap, in the PSO algorithm for improving solution quality. The proposed methodology was tested against the existing solutions on some real problem instances. They demonstrated that the PSO-based algorithm yielded better results from the existing solutions for all the test problems.

Truong and Kachitvichyanukul (2007) proposed a solution methodology that hybridised PSO with GA. Their results on standard benchmark problems outperformed several heuristic procedures and were able to deal with bigger size problems than the benchmarks. For solving the same problem, Jarboui et al. (2008) proposed a variant of PSO that they called combinatorial PSO. Their results over standard set of problem instances outperformed the simulated annealing algorithm results.

Chen et al. (2006) has applied PSO for solving the CVRP. They were using PSO with discrete value of particle's position and hybridised it with simulated annealing. Their application performed well on small-size benchmark problems. Continuous PSO algorithm has also been proposed and successfully applied for solving CVRP (Ai and Kachitvichyanukul, 2007). Qing et al. (2006) applied PSO for VRPTW; however, the effectiveness of PSO on this problem has not been well proven since it is only demonstrated for solving a very small problem case.

From the review, it can be observed that the major aspect of PSO application to specific problem is the definition of particle position and its transformation method to the problem solution. Various definitions have been proposed in the literature with their own characteristics and performances. Sometimes, the PSO algorithm is modified and hybridised with other methods for enhancing the algorithm performance. Also, embedding special heuristic for the respective problem into PSO algorithm is a common mode to improve the performance of the algorithm.

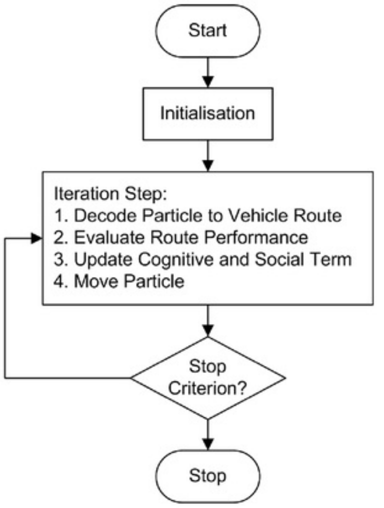
3 PSO framework for solving VRPTW

As mentioned earlier, a PSO algorithm starts by initialising a swarm of particles followed by a series of iteration steps to move the particles towards the best position. This section will explain in details the proposed PSO framework for solving VRPTW.

The PSO framework for solving VRPTW is proposed based on the GLNPSO, a PSO Algorithm with multiple social learning structures (Pongchairerks and Kachitvichyanukul, 2005). This version of PSO incorporates two additional terms to express social behaviour. The first term is called local best, which is defined as the position of a particle that has the best fitness value among several adjacent particles. The second term is called near neighbour best, which is determined based on the Fitness-Distance-Ratio (*FDR*), a relative measurement of fitness value and position between position of a particle and personal best of other particles.

It is noted that PSO works on finding the best position of particles. So, to apply PSO to a specific problem, i.e., VRPTW, the relationship between the position particles and the problem solution, i.e., vehicle route, must be clearly defined. A method for converting particle position to problem solution is usually called '*decoding method*'. The PSO framework for solving VRPTW is described in Figure 1 and the details of each step and each component will be described in the following sub-sections.

Figure 1 PSO framework for solving VRPTW


Start
Initialisation
Iteration Step:
1. Decode Particle to Vehicle Route
2. Evaluate Route Performance
3. Update Cognitive and Social Term
4. Move Particle
Stop Criterion?
Stop

3.1 Initialisation

In the initialisation step, a swarm consisting of I particles is initialised by setting the position, velocity, and personal best value of each particle. For each dimension of a particle, the position value (x_{id}) is set randomly in the range of minimum and maximum position value, $x_{id} = U(x^{\min}, x^{\max})$, the velocity value is initially defined as zero, $v_{id} = 0$, and personal best value (p_{id}) is equal to the position value, $p_{id} = x_{id}$.

3.2 Iteration step

The main part of PSO framework is the iteration step. As shown in Figure 1, each iteration step consists of four sub-steps: decode particle to vehicle route, evaluate route performance, update cognitive and social term, and update particle velocity and position. In the first sub-step, each particle position is decoded to a set of vehicle routes by the decoding method that will be explained in detail in Section 3. Then, the performance of each constructed set of routes, i.e., the total travelled distance of the routes, is evaluated. This performance value, then, is kept as the fitness value of its corresponding particle. For example, after position of particle $i(\boldsymbol{X}_i)$ is decoded to the set of vehicle routes $\boldsymbol{R}_i$, the total travelled distance of the routes $\boldsymbol{R}_i$, denoted by $\varphi(\boldsymbol{X}_i)$, is kept as the fitness value of $\boldsymbol{X}_i$.

After the fitness value of every particle is determined, the information of the cognitive and social terms of each particle, which are personal best (*pbest*), global best (*gbest*), local best (*lbest*), and near neighbour best (*nbest*) are updated. It is noted that smaller fitness value is desirable, since the VRPTW objective is to minimise total travelled distance. The updating procedure is explained as follows. First, the fitness value of each particle, $\varphi(\boldsymbol{X}_i)$, is compared against its *pbest*, $\varphi(\boldsymbol{P}_i)$. The *pbest* is set to be the current position, $\boldsymbol{P}_i = \boldsymbol{X}_i$, if the fitness value of current position is smaller than its *pbest*, $\varphi(\boldsymbol{X}_i) < \varphi(\boldsymbol{P}_i)$. Also, the *gbest* is set to be the current position, $\boldsymbol{P}_g = \boldsymbol{X}_i$, if the fitness value of current position is smaller than its *gbest*, $\varphi(\boldsymbol{X}_i) < \varphi(\boldsymbol{P}_g)$. Second, find the smallest *pbest* fitness value among *K* immediate neighbour of each particle and set the corresponding *pbest* as the *lbest*,

$$\boldsymbol{P}_i^L = \min_{j \in N_i}(\boldsymbol{P}_j),$$

where N_i is the set of immediate neighbour of particle *i*. Finally, each dimension of particle *nbest* is determined as the *pbest* of other particles at corresponding dimension $(p_{id}^N = p_{jd})$ that maximise *FDR*, where

$$\text{FDR} = \frac{\varphi(\boldsymbol{X}_i) - \varphi(\boldsymbol{P}_j)}{|x_{id} - p_{jd}|} \quad \text{where } i \neq j. \tag{1}$$

In the last sub-step, the velocity of particle is updated based on the social and cognitive terms before the particle is moved to a new position based on its velocity according to the following equations:

$$v_{id} = wv_{id} + c_p u(p_{id} - x_{id}) + c_g u(p_{gd} - x_{id}) + c_l u(p_{id}^L - x_{id}) + c_n u(p_{id}^N - x_{id}) \tag{2}$$

$$x_{id} = x_{id} + v_{id} \tag{3}$$

where *u* is a uniform distributed random number within interval [0,1]; *w* is the inertia weight; c_p, c_g, c_l, and c_n are the acceleration constant of cognitive term, *gbest* social term, *lbest* social term and *nbest* social term, respectively. For controlling the search range of particles, if the position of a particle in a particular dimension exceeds the minimum or maximum position value ($x_{id} < x^{\min}$ or $x_{id} > x^{\max}$), then the position value is set on the bound ($x_{id} = x^{\min}$ or $x_{id} = x^{\max}$) and the corresponding velocity value is set to zero ($v_{id} = 0$).

3.3 *Stop criterion*

This step will control the mechanism for repetition of this algorithm. The algorithm is terminated whenever the stopping criterion is satisfied; otherwise, the iteration step will be repeated. The stopping criterion used in this framework is the total number of iteration.

4 Solution representation and decoding method

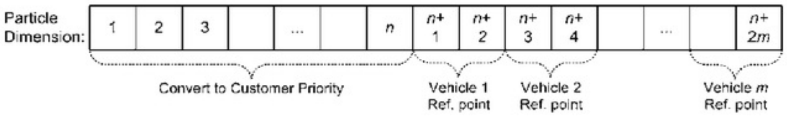
The description of the particle that is used in a specific PSO application is usually called the solution representation and it must be clearly defined. By a decoding method, a solution representation can be transformed to a specific problem solution, i.e., a set of vehicle routes for VRPTW. Both the solution representation and decoding method must be carefully designed to achieve an effective application.

The decoding method is the first part of the iteration step in the PSO framework for solving VRPTW described in Section 3. The decoding method is responsible for transforming each specific particle representation into a set of vehicle routes in which its performance will be kept as the fitness value of the particle. The proposed solution representation and decoding method for this framework are explained in the following sub-sections.

4.1 Solution representation

An indirect representation of $(n+2m)$-dimensional particle is proposed to represent VRPTW solution with n customers and m vehicles (see Figure 2). Particle position in each dimension is represented by a real number, so that it can be applied in the framework based on a continuous version of PSO. The indirect representation is chosen since the vehicle routes cannot be directly represented by continuous numbers. This representation can be transformed to vehicle routes by a special decoding method explained in the next sub-section.

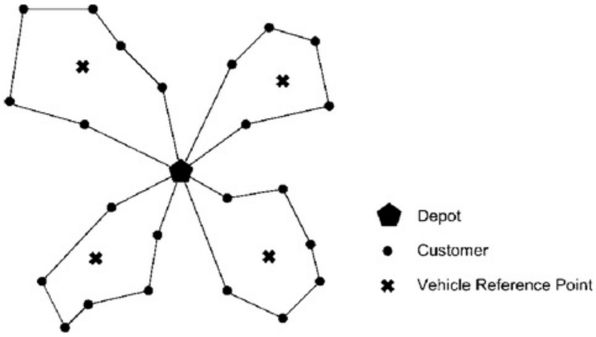
Figure 2 Solution representation



The proposed solution representation consists of two main parts. The first part of the representation is related to the customers, in which each customer is represented by a factor that can be used to set the priority of the customers. Hence, n dimensions of a particle are required to represent n customers. In the decoding step, a customer who has higher priority will be inserted to the vehicle routes before the one with lower priority. The advantage of this representation is that a vast variety of priority lists could be generated. Since it is possible to get infinite array combination of n continuous numbers and one combination leads to one priority list, a diverse set of vehicle routes can be generated. It is expected that the PSO mechanism can move its particles into diverse position during its iteration process, so that diverse sets of vehicle routes can be evaluated during the algorithm run. It will increase the possibility to find a better final solution.

The second part of the representation is related to the vehicles that incorporate the unique idea of vehicle reference point. Reference point of a vehicle is defined as a point in the service map that represents a certain area that is most likely be served by the vehicle. As a consequence, a vehicle route tends to aggregate around its corresponding reference point (see Figure 3) and yield a shorter travelled distance. Therefore, two dimensions of a particle are required for each vehicle, one for the x-position and the other for the y-position of the reference point. Hence, a total of $2m$ dimensions are needed to represent m vehicles. In the decoding step, these reference points are used for setting the vehicle assignment priority of each customer. The advantage of this representation is the tendency to construct routes with shorter distance by using the vehicle reference points. In addition, various combinations of vehicle reference points can be generated by PSO mechanism, which leads to more diverse sets of vehicle routes to be constructed.

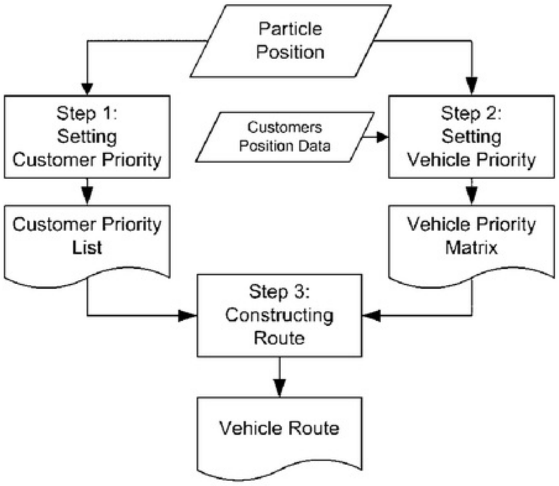
Figure 3 Vehicle routes and its reference point



4.2 Decoding method

The basic mechanism of the proposed decoding method, which consists of three steps for converting the proposed solution representation into VRPTW solution, is illustrated in Figure 4. The first step of the decoding method is to set a priority list of customers based on the corresponding dimensions of particle's position. After that, a vehicle priority matrix is constructed based on the vehicle reference points, which are extracted from the corresponding dimensions of particle position, and the information on customers' location. Finally, a set of vehicle routes is generated based on the customer priority list and the vehicle priority matrix. The formal definition of the proposed decoding method is described in the following algorithm.

Figure 4 Decoding steps of particle position into vehicle route



Decoding Algorithm:

1 Setting Customer Priority List

  a Take out the first n dimensions of position value as the corresponding position value of customers.

  b Sort the customer index based on its corresponding position value in ascending order.

  c Take the sorted *customer* index as the customer priority list.

2 Setting Vehicle Priority Matrix

  a Take out the $2m$ dimensions of position value as the vehicle route reference points.

  b For each customer in the customer priority list:

    i Calculate the Euclidean distance between the customer and vehicle route reference points.

    ii Sort the vehicle index based on its Euclidean distance in ascending order.

    iii Take the sorted vehicle index as the corresponding row for the customer in the vehicle priority matrix.

3 Route Construction

  For each customer in the customer priority list, starting from the first to the last priority:

  a Set j as the first vehicle priority of the customer.

  b Make a new candidate route by inserting the customer to the position that has the smallest additional cost in route j.

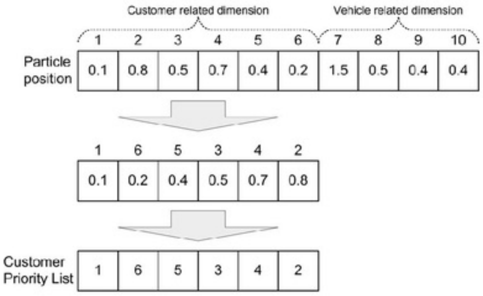
  c Check feasibility of the candidate route by evaluating all constraints: vehicle capacity, working time, and time window constraints.

  d If a feasible solution is reached, update route j with the candidate route and re-optimise emerging route with 2-opt method; then return to step 3a with the next customer.

  e If the candidate route is infeasible, set j as the next vehicle priority of the customer; then go to step 3b.

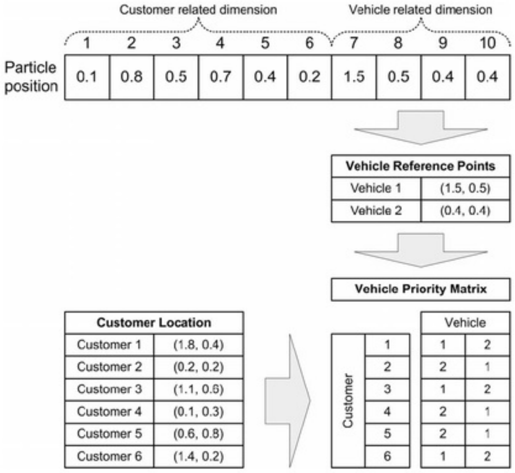
The first step of decoding method is setting a priority list of customer based on the corresponding n dimensions of particle position. The priority list is constructed using rule of small position value, in which the smaller the position value of a dimension, the higher is the priority given to the corresponding customer. As an illustration, consider a problem with 6 customers and 2 vehicles. Hence, a ten-dimensional particle is needed to represent a solution of this problem. The conversion process of a typical position value to its corresponding customer priority list is illustrated in Figure 5.

Figure 5 Setting customer priority list



In the second step, the corresponding $2m$ dimensions are converted into reference point of vehicles. Then, the vehicle priority matrix is constructed based on the relative distance between these points and customer locations in which a customer is prioritised to be served by vehicle with closer distance. It is noted that the distance can be calculated as long as the reference points and the customer locations are placed in the same Cartesian map. An illustration of this step with the same typical particle's position value is shown in Figure 6.

Figure 6 Setting vehicle priority matrix



Vehicle Reference Points	
Vehicle 1	(1.5, 0.5)
Vehicle 2	(0.4, 0.4)

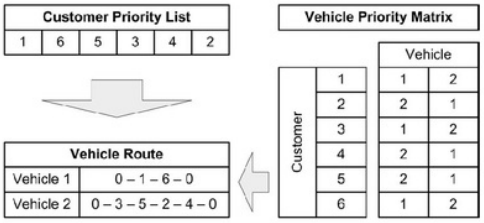
Customer Location	
Customer 1	(1.8, 0.4)
Customer 2	(0.2, 0.2)
Customer 3	(1.1, 0.6)
Customer 4	(0.1, 0.3)
Customer 5	(0.6, 0.8)
Customer 6	(1.4, 0.2)

Vehicle Priority Matrix

		Vehicle	
Customer	1	1	2
	2	2	1
	3	1	2
	4	2	1
	5	2	1
	6	1	2

The last decoding step is to construct routes 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 other problem constraints, such as vehicle capacity constraint, vehicle working time constraint, and customer time window constraint. This newly assigned customer is inserted according to the cheapest insertion heuristic, i.e., it is inserted to the best sequence in the existing vehicle route based on the least additional cost. Another effort to improve solution quality of the route is to re-optimise the emerging route by using some improvement heuristic methods, i.e., 2-opt method. By these two embedded heuristics, solution quality of the constructed route, i.e., the travelled distance, is maintained as high as possible. An illustration of route construction step based on the same example as above (mentioned in Figures 5 and 6) is given in Figure 7.

Figure 7 Route construction



Customer Priority List					
1	6	5	3	4	2

Vehicle Priority Matrix

	Vehicle	
Customer		
1	1	2
2	2	1
3	1	2
4	2	1
5	2	1
6	1	2

Vehicle Route	
Vehicle 1	0 – 1 – 6 – 0
Vehicle 2	0 – 3 – 5 – 2 – 4 – 0

5 Computational experiment

Computational experiments are conducted to evaluate the effectiveness of the proposed method using benchmark data of Solomon (1987) as the test cases. 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. The PSO parameters are set similar with the previous work of Ai and Kachitvichyanukul (2007). The PSO parameters are listed in Table 1. For each case, five independent replications of the proposed method are performed.

Table 1 Parameter setting of PSO algorithm

Parameter	*Value*
Number of particle (I)	100
Number of iteration (T)	1000
Number of neighbour (K)	5
Inertia weight (w)	Linearly decreasing from 0.9 to 0.4
Personal best position acceleration constant (c_p)	0.5
Global best position acceleration constant (c_g)	0.5
Local best position acceleration constant (c_l)	1.5
Near neighbour best position acceleration constant (c_n)	1.5

It is noted that the VRPTW benchmark data of Solomon comprises of high variety of problem situations, such as location of customers, vehicles characteristic, customers' demand and corresponding time windows, in which the variation is implied on the problem instance name. In terms of customers' location, there are problems with clustered customer (instances Cxxx), problems with dispersed customer (instances Rxxx), and problems with combination of clustered and dispersed customer (instances RCxxx). In terms of vehicle characteristic, there are problems with low capacity vehicles (instances x1xx) and problems with high capacity vehicles (instances x2xx). Two last digits of instance number exemplify the difference in customers' demand and time windows. Therefore, testing the proposed method on this set of benchmark problems is a suitable way to demonstrate the generality of the method.

To assess the performance of the proposed method, two criteria are included: solution quality and computational time. If the proposed method is able to find solutions that approach optimal solution within short computational time, it can be stated as an effective method. In terms of solution quality, the solution obtained from PSO is compared against the corresponding optimal solution, which can be accessed from Solomon's website (http://web.cba.neu.edu/~msolomon/problems.htm). The percentage deviation from the optimal solution will be used to measure solution quality.

The computational test results for problems with 25 customers and 50 customers are shown in Tables 2 and 3, respectively. The statistics include the minimum, average and maximum value of total travelled distance and computational time. The problems are solved five times and the results are presented along with deviation from optimal solution and its corresponding computational time.

Table 2 Solution summary for 25-customer problems

Problem instance	*No. of vehicle*	*Optimal solution*	*PSO solution*			*Deviation from optimal solution*			*Computational time* (second)		
			Min	*Ave*	*Max*	*Min* (%)	*Ave* (%)	*Max* (%)	*Min*	*Ave*	*Max*
C101	3	191.3	191.8	191.8	191.8	0.3	0.3	0.3	35	37	41
C102	3	190.3	190.7	190.7	190.7	0.2	0.2	0.2	33	34	35
C103	3	190.3	190.7	190.7	190.7	0.2	0.2	0.2	33	35	36
C104	3	186.9	187.4	192.1	200.9	0.3	2.8	7.5	33	34	34
C105	3	191.3	191.8	191.8	191.8	0.3	0.3	0.3	35	38	41
C106	3	191.3	191.8	191.8	191.8	0.3	0.3	0.3	38	40	43
C107	3	191.3	191.8	191.8	191.8	0.3	0.3	0.3	36	37	39
C108	3	191.3	191.8	191.8	191.8	0.3	0.3	0.3	36	38	40
C109	3	191.3	191.8	191.8	191.8	0.3	0.3	0.3	35	38	42
C201	2	214.7	215.5	215.5	215.5	0.4	0.4	0.4	56	60	63
C202	2	214.7	215.5	215.5	215.5	0.4	0.4	0.4	49	53	60
C203	2	214.7	215.5	215.5	215.5	0.4	0.4	0.4	50	54	58
C204	2	213.1	213.9	213.9	213.9	0.4	0.4	0.4	87	91	94
C205	2	214.7	215.5	215.5	215.5	0.4	0.4	0.4	52	59	67
C206	2	214.7	215.5	215.5	215.5	0.4	0.4	0.4	49	53	58
C207	2	214.5	215.3	215.3	215.3	0.4	0.4	0.4	52	53	55
C208	2	214.5	215.4	215.4	215.4	0.4	0.4	0.4	49	50	50
R101	8	617.1	618.3	618.3	618.3	0.2	0.2	0.2	33	33	34
R102	7	547.1	548.1	548.1	548.1	0.2	0.2	0.2	30	31	31
R103	5	454.6	455.7	459.2	473.2	0.2	1.0	4.1	28	29	29
R104	4	416.9	418.0	418.1	418.3	0.3	0.3	0.3	28	29	30
R105	6	530.5	531.5	531.5	531.5	0.2	0.2	0.2	30	30	30

Table 2 Solution summary for 25-customer problems (continued)

			PSO solution			*Deviation from optimal solution*			*Computational time* (second)		
Problem instance	*No. of vehicle*	*Optimal solution*	*Min*	*Ave*	*Max*	*Min* (%)	*Ave* (%)	*Max* (%)	*Min*	*Ave*	*Max*
R106	3	465.4	466.5	466.5	466.5	0.2	0.2	0.2	29	30	32
R107	4	424.3	425.3	426.8	429.2	0.2	0.6	1.2	29	29	30
R108	4	397.3	398.3	398.3	398.3	0.3	0.3	0.3	27	28	28
R109	5	441.3	442.6	442.6	442.6	0.3	0.3	0.3	28	29	29
R110	4	444.1	445.9	449.9	452.8	0.4	1.3	1.9	30	31	32
R111	5	428.8	429.7	432.9	441.7	0.2	1.0	3.0	29	30	30
R112	4	393.0	394.1	394.1	394.1	0.3	0.3	0.3	27	28	28
R201	4	463.3	464.4	464.4	464.4	0.2	0.2	0.2	29	30	31
R202	4	410.5	411.5	411.5	411.5	0.2	0.2	0.2	29	30	33
R203	3	391.4	392.3	392.3	392.3	0.2	0.2	0.2	30	31	32
R204	2	355.0	358.6	358.6	358.6	1.0	1.0	1.0	37	38	40
R205	3	393.0	395.8	395.8	395.8	0.7	0.7	0.7	30	32	33
R206	3	374.4	378.8	378.8	378.8	1.2	1.2	1.2	32	33	33
R207	3	361.6	362.6	362.6	362.6	0.3	0.3	0.3	30	30	32
R208	1	328.2	329.3	329.3	329.3	0.3	0.3	0.3	89	93	99
R209	2	370.7	371.6	371.6	371.6	0.2	0.2	0.2	45	45	46
R210	3	404.6	405.5	405.5	405.5	0.2	0.2	0.2	33	33	34
R211	2	350.9	353.8	353.8	353.8	0.8	0.8	0.8	37	39	41
RC101	4	461.1	462.2	462.2	462.2	0.2	0.2	0.2	36	37	39
RC102	3	351.8	352.7	352.7	352.7	0.3	0.3	0.3	36	37	38
RC103	3	332.8	333.9	333.9	333.9	0.3	0.3	0.3	32	34	39
RC104	3	306.6	307.1	307.1	307.1	0.2	0.2	0.2	30	31	33
RC105	4	411.3	412.4	412.4	412.4	0.3	0.3	0.3	34	35	37
RC106	3	345.5	346.5	347.0	347.3	0.3	0.4	0.5	34	36	39
RC107	3	298.3	298.9	298.9	298.9	0.2	0.2	0.2	30	31	31
RC108	3	294.5	295.0	295.0	295.0	0.2	0.2	0.2	29	30	31
RC201	3	360.2	361.2	361.2	361.2	0.3	0.3	0.3	45	48	52
RC202	3	338.0	338.8	338.8	338.8	0.2	0.2	0.2	40	41	43
RC203	3	326.9	327.7	327.7	327.7	0.2	0.2	0.2	36	38	41
RC204	3	299.7	300.2	300.2	300.2	0.2	0.2	0.2	33	34	35
RC205	3	338.0	338.9	338.9	338.9	0.3	0.3	0.3	41	43	45
RC206	3	324.0	325.1	325.1	325.1	0.3	0.3	0.3	40	43	47
RC207	3	298.3	298.9	298.9	298.9	0.2	0.2	0.2	34	35	38
RC208	2	269.1	269.6	269.6	269.6	0.2	0.2	0.2	48	49	50

Table 3 Solution summary for 50-customer problems

Problem instance	*No. of vehicle*	*Best Known solution*	*PSO solution*			*Deviation from best known solution*			*Computational time* (second)		
			Min	*Ave*	*Max*	*Min* (%)	*Ave* (%)	*Max* (%)	*Min*	*Ave*	*Max*
C101	5	362.4	363.2	377.5	404.6	0.2	4.2	11.7	89	102	116
C102	5	361.4	373.5	408.9	432.6	3.4	13.1	19.7	99	106	114
C103	5	361.4	387.4	430.9	543.7	7.2	19.2	50.4	79	98	122
C104	5	358.0	366.7	389.1	412.2	2.4	8.7	15.1	74	88	100
C105	5	362.4	363.2	383.9	430.9	0.2	5.9	18.9	88	95	105
C106	5	362.4	363.2	370.3	398.4	0.2	2.2	9.9	88	97	111
C107	5	362.4	363.2	406.5	443.6	0.2	12.2	22.4	82	90	98
C108	5	362.4	363.2	403.4	479.8	0.2	11.3	32.4	84	93	100
C109	5	362.4	385.4	420.1	463.5	6.3	15.9	27.9	82	88	100
C201	3	360.2	361.8	361.8	361.8	0.4	0.4	0.4	179	184	189
C202	3	360.2	361.8	361.8	361.8	0.4	0.4	0.4	173	179	186
C203	3	359.8	361.4	361.4	361.4	0.4	0.4	0.4	176	179	185
C204	2	350.1	351.7	351.7	351.7	0.5	0.5	0.5	183	191	199
C205	3	359.8	361.4	361.4	361.4	0.4	0.4	0.4	180	184	189
C206	3	359.8	361.4	361.4	361.4	0.4	0.4	0.4	178	185	192
C207	3	359.6	361.2	361.2	361.2	0.4	0.4	0.4	173	181	196
C208	2	350.5	352.1	352.1	352.1	0.5	0.5	0.5	190	196	202
R101	12	1044.0	1053.9	1054.3	1055.0	0.9	1.0	1.1	86	87	90
R102	11	909.0	913.6	918.8	931.7	0.5	1.1	2.5	79	82	85
R103	9	772.9	778.5	780.8	785.2	0.7	1.0	1.6	75	76	76
R104	6	625.4	632.2	634.9	639.1	1.1	1.5	2.2	72	72	73
R105	9	899.3	932.9	945.0	968.0	3.7	5.1	7.6	79	81	84
R106	9	793.0	797.3	810.3	817.6	0.5	2.2	3.1	75	78	79
R107	7	711.1	713.9	732.7	744.3	0.4	3.0	4.7	76	77	78
R108	6	617.7	620.3	620.7	621.6	0.4	0.5	0.6	70	72	74
R109	8	786.8	803.8	817.2	830.8	2.2	3.9	5.6	75	76	77
R110	7	697.0	708.4	731.8	742.0	1.6	5.0	6.5	75	77	79
R111	7	707.2	724.2	728.0	732.8	2.4	2.9	3.6	74	75	76
R112	6	630.2	637.8	642.6	646.0	1.2	2.0	2.5	74	76	77
R201	6	791.9	815.4	815.7	817.1	3.0	3.0	3.2	110	115	118
R202	5	698.5	722.2	725.9	729.5	3.4	3.9	4.4	99	103	108
R203	5	605.3	613.7	616.6	618.4	1.4	1.9	2.2	81	84	87
R204	2	506.4	507.1	507.1	507.1	0.1	0.1	0.1	163	168	174
R205	4	690.1	706.4	709.0	713.9	2.4	2.7	3.4	107	114	121

Table 3 Solution summary for 50-customer problems (continued)

Problem instance	*No. of vehicle*	*Best known solution*	*PSO solution*			*Deviation from best known solution*			*Computational time* (second)		
			Min	*Ave*	*Max*	*Min* (%)	*Ave* (%)	*Max* (%)	*Min*	*Ave*	*Max*
R206	4	632.4	638.6	638.6	638.6	1.0	1.0	1.0	92	98	101
R209	4	600.6	624.6	624.6	624.6	4.0	4.0	4.0	92	93	94
R210	4	645.6	649.9	657.8	662.7	0.7	1.9	2.6	99	101	103
R211	3	535.5	538.7	539.2	541.3	0.6	0.7	1.1	107	111	116
RC101	8	944.0	945.6	947.9	952.9	0.2	0.4	0.9	87	89	92
RC102	7	822.5	828.0	830.3	835.6	0.7	0.9	1.6	84	88	90
RC103	6	710.9	712.6	713.1	715.3	0.2	0.3	0.6	82	84	88
RC104	5	545.8	546.5	546.5	546.5	0.1	0.1	0.1	73	74	75
RC105	8	855.3	857.7	860.9	865.4	0.3	0.7	1.2	81	86	89
RC106	6	723.2	757.2	761.6	768.0	4.7	5.3	6.2	86	88	89
RC107	6	642.7	645.4	645.5	645.6	0.4	0.4	0.4	79	80	82
RC108	6	598.1	599.2	599.2	599.2	0.2	0.2	0.2	74	75	77
RC201	5	684.8	686.3	686.3	686.3	0.2	0.2	0.2	109	114	125
RC202	5	613.6	615.0	615.0	615.0	0.2	0.2	0.2	90	95	108
RC203	4	555.3	556.5	556.5	556.5	0.2	0.2	0.2	107	111	114
RC204	3	444.2	445.0	445.0	445.0	0.2	0.2	0.2	107	109	110
RC205	5	630.2	632.0	632.0	632.0	0.3	0.3	0.3	97	102	105
RC206	5	610.0	611.7	611.7	611.7	0.3	0.3	0.3	89	93	96
RC207	4	558.6	559.9	559.9	559.9	0.2	0.2	0.2	119	123	124

As it is shown in Table 2, the solution quality of PSO for 25-customer problems is good and consistent. Overall, the average results are very close to the optimal solution as indicated by the average deviation that is less than 1%. Moreover, consistency over replications is implied by the results. Almost all results give the same result over five replications as shown by the same value of minimum, average and maximum value of the objective function. Exceptions do appear for the cases C104, R103, R107, R110, R111, and RC106; however, the variations are small as shown by the narrow ranges between maximum and minimum value of the objective function, and also for the ranges of the objective function over five replications. Since these results came from various problem situations, the results demonstrated that the proposed method yields a reasonably good result for the 25-customer problems in general.

It is also shown from Table 2 that the proposed method yields good solutions with reasonable fast computational times. For the 25-customer problems, almost all the instances are solved within a minute with the exceptions of cases C204 and R208, which required 1.5 min. In addition, no large variations were observed in the computational time over five replications for each problem instance. This result is also desirable, since a good quality of solution is guaranteed to be reached in reasonable amount of time.

As it is shown in Table 3, the majority of PSO results are also good and consistent for 50-customer problems. The average results are also very close to the optimal solution as demonstrated by a small average deviation. The consistency over replications is shown again by a narrow range between minimum and maximum value of objective function. However, there are some cases in which the average deviation is higher than 1% with slightly larger variations over replications.

There are also a few exceptions for some 50-customer problems, i.e., instances C102, C103, C107, C108, and C109, which showed a high deviation and variation in the results. Since this fact might be caused by the construction heuristic, the solution representation, or the problem case characteristics, further exploration on this area is still required to further the performance of the proposed method.

It is also shown in Table 3 that most of the 50-customer problems required less than 2 min of computational time. Some exceptions did appear for cases C2xx and R204, in which over 3 min of computational time is required. However, the solution quality of these exception cases remains very good and consistent. Additionally, the computational time over five replications for solving each instance is also showing narrow variation that is similar with the result of 25-customer problems.

The results on computational time also implied that the average computational time for a 50-customer problem, which is less than 2 min, is approximately twice of its 25-customer problem, in which most of the 25-customer problems are solved within a minute. In other words, it is empirically shown that the computational time of this proposed method is linearly proportional with the problem size, i.e., total number of customers. This attribute is usually preferred by practitioners who must regularly deal with large problems, since they could manage the computational time required to solve a particular problem.

6 Conclusion

The computational experiments show that the proposed PSO framework and solution representation is effective to solve the VRPTW. The effectiveness of the proposed method comes from the idea of vehicle reference point, route construction heuristics, and the simplicity of the PSO. The vehicle reference point ensures that the constructed route will cover only a narrow area. The route construction heuristics is capable of increasing the solution quality of the route. Also, the structure and mechanism of PSO can generate diverse solutions and consistently maintain or improve the best found solution. Moreover, since the time performance of the proposed PSO is relatively short and linearly proportional with the problem size, the proposed PSO will be useful for the practitioners who usually deal with large problems.

Further research to improve the performance of this algorithm is needed, especially to deal with high deviation and variation in the result of some problem cases. Application of the proposed method to other VRP variants or type of problem is also promising. Since the variants of VRP differ from one another only on the specific problem constraints, any adjustments needed can be made at the constraint feasibility checking in the decoding method. However, the effectiveness of this idea remains to be further evaluated.

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 thank AITHPC and the Thai GRID Center for the access of computing facility and the technical support. The authors also express thanks to the anonymous reviewers for the valuable comments and suggestions on the preparation of this paper.

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

Ai, T.J. and Kachitvichyanukul, V. (2007) 'A particle swarm optimization for the capacitated vehicle routing problem', *International Journal of Logistics and SCM Systems*, Vol. 2, No. 1, pp.50–55.

Akjiratikarl, C., Yenradee, P. and Drake, P.R. (2007) 'PSO-based algorithm for home care worker scheduling in the UK', *Computers and Industrial Engineering*, Vol. 53, No. 4, pp.559–583.

Alvarenga, G.B., Mateus, G.R. and de Tomi, G. (2007) 'A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows', *Computers and Operations Research*, Vol. 34, No. 6, pp.1561–1584.

Bard, J.F., Kontoravdis, G. and Yu, G. (2002) 'A branch-and-cut procedure for the vehicle routing problem with time windows', *Transportation Science*, Vol. 36, No. 2, pp.250–269.

Berger, J. and Barkaoui, M. (2004) 'A parallel hybrid genetic algorithm for the vehicle routing problem with time windows', *Computers and Operations Research*, Vol. 31, No. 12, pp.2037–2053.

Bräysy, O. and Gendreau, M. (2005a) 'Vehicle routing problem with time windows, part I: route construction and local search algorithms', *Transportation Science*, Vol. 39, No. 1, pp.104–118.

Bräysy, O. and Gendreau, M. (2005b) 'Vehicle routing problem with time windows, part II: metaheuristics', *Transportation Science*, Vol. 39, No. 1, pp.119–139.

Chen, A.L., Yang, G.K. and Wu, Z.M. (2006) 'Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem', *Journal of Zhejiang University: Science*, Vol. 7, No. 4, pp.607–614.

Clerc, M. (2006) *Particle Swarm Optimization*, ISTE, London.

Gambardella, L.M., Taillard, E. and Agazzi, G. (1999) 'MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows', in Corne, D., Dorigo, M. and Glover, F. (Eds.): *New Ideas in Optimization*, McGraw-Hill, London, pp.63–76.

Jarboui, B., Damak, N., Siarry, P. and Rebai, A. (2008) 'A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems', *Applied Mathematics and Computation*, Vol. 195, No. 1, pp.299–308.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, pp.1942–1948.

Kennedy, J. and Eberhart, R.C. (2001) *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco.

Kohl, N. and Madsen, O.B.G. (1997) 'An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation', *Operations Research*, Vol. 45, No. 3, pp.395–406.

Pongchairerks, P. and Kachitvichyanukul, V. (2005) 'A non-homogenous particle swarm optimization with multiple social structures', *Proceedings of the 2005 International Conference on Simulation and Modeling*, Nakorn Pathom, Thailand, pp.132–136.

Pongchairerks, P. and Kachitvichyanukul, V. (2007) 'A comparison between algorithms VNS with PSO and VNS without PSO for job-shop scheduling problems', *International Journal of Computational Science*, Vol. 1, No. 2, pp.179–191.

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

Pratchayaborirak, T. and Kachitvichyanukul, V. (2007) 'A two-stage particle swarm optimization for multi-objective job shop scheduling problems', *Proceedings of the 8th Asia Pacific Industrial Engineering and Management System Conference (APIEMS 2007)*, Paper No. 569, Kaohsiung, Taiwan.

Qing, Z., Limin, Q., Yingchun, L. and Shanjun, Z. (2006) 'An improved particle swarm optimization algorithm for vehicle routing problem with time windows', *Proceedings of 2006 IEEE Congress on Evolutionary Computation (CEC 2006)*, Vancouver, Canada, pp.1386–1390.

Sha, D.Y. and Hsu, C.Y. (2006) 'A hybrid particle swarm optimization for job shop scheduling problem', *Computers and Industrial Engineering*, Vol. 51, No. 4, pp.791–808.

Solomon, M.M. (1987) 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Operations Research*, Vol. 35, No. 2, pp.254–265.

Tan, K.C., Lee, L.H. and Ou, K. (2001) 'Hybrid genetic algorithms in solving vehicle routing problems with time window constraints', *Asia-Pacific Journal of Operational Research*, Vol. 18, No. 1, pp.121–130.

Tasgetiren, M.F., Liang, Y.C., Sevkli, M. and Gencyilmaz, G. (2007) 'A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem', *European Journal of Operational Research*, Vol. 177, No. 3, pp.1930–1947.

Truong, V.X. and Kachitvichyanukul, V. (2007) 'A hybrid PSO algorithm for multi-mode resource-constrained project scheduling problems', *Proceedings of the 8th Asia Pacific Industrial Engineering and Management System Conference (APIEMS 2007)*, Paper No. 618, Kaohsiung, Taiwan.

Paper 03

ORIGINALITY REPORT

43%	35%	24%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.arpapress.com Internet Source	19%
2	The Jin Ai, Voratas Kachitvichyanukul. "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery", Computers & Operations Research, 2009 Publication	4%
3	iieom.org Internet Source	3%
4	www.inderscience.com Internet Source	2%
5	docplayer.net Internet Source	2%
6	Submitted to Asian Institute of Technology Student Paper	1%
7	www.inderscienceonline.com Internet Source	1%
	Submitted to Beykoz Lojistik Meslek	

8	Yüksekokulu Student Paper	1%
9	Liu, Jie, and Voratas Kachitvichyanukul. "A particle swarm optimisation algorithm for the capacitated location-routing problem", International Journal of Operational Research, 2015. Publication	1%
10	Akjiratikarl, C.. "PSO-based algorithm for home care worker scheduling in the UK", Computers & Industrial Engineering, 200711 Publication	1%
11	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	<1%
12	en.digitalbitrate.com Internet Source	<1%
13	Submitted to Erasmus University of Rotterdam Student Paper	<1%
14	link.springer.com Internet Source	<1%
15	forums.worden.com Internet Source	<1%
16	The Jin Ai, Voratas Kachitvichyanukul. "Particle swarm optimization and two solution	<1%

	representations for solving the capacitated vehicle routing problem", Computers & Industrial Engineering, 2009 Publication	
17	Ai, T.J.. "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery", Computers and Operations Research, 200905 Publication	<1%
18	www.tc.ait.ac.th Internet Source	<1%
19	Hai Shen. "Particle Swarm Optimization in Solving Vehicle Routing Problem", 2009 Second International Conference on Intelligent Computation Technology and Automation, 10/2009 Publication	<1%
20	art2you.co.kr Internet Source	<1%
21	uhdspace.uhasselt.be Internet Source	<1%
22	www.idsia.ch Internet Source	<1%
23	Submitted to University of Witwatersrand Student Paper	<1%

24	research.sabanciuniv.edu Internet Source	<1%
25	Ai, T.J.. "Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem", Computers & Industrial Engineering, 200902 Publication	<1%
26	EcoProduction, 2015. Publication	<1%
27	140.118.201.168 Internet Source	<1%
28	Submitted to The University of Nottingham, Malaysia Student Paper	<1%
29	scholarbank.nus.edu.sg Internet Source	<1%
30	arc.aiaa.org Internet Source	<1%
31	www.koreascience.or.kr Internet Source	<1%
32	tci-thaijo.org Internet Source	<1%
33	ostfeld.net.technion.ac.il Internet Source	<1%

34	apiems.net Internet Source	<1%
35	Byung-In Kim, So-Jung Son. "A probability matrix based particle swarm optimization for the capacitated vehicle routing problem", Journal of Intelligent Manufacturing, 2010 Publication	<1%
36	Nguyen, Su, and Voratas Kachitvichyanukul. "An efficient differential evolution algorithm for multi-mode resource-constrained project scheduling problems", International Journal of Operational Research, 2012. Publication	<1%
37	epdf.tips Internet Source	<1%
38	etheses.whiterose.ac.uk Internet Source	<1%
39	mafiadoc.com Internet Source	<1%
40	www.cs.uoi.gr Internet Source	<1%
41	e-space.mmu.ac.uk Internet Source	<1%
42	eprints.soton.ac.uk Internet Source	<1%

43	Vincent F. Yu, A. A. N. Perwira Redi, Chao-Lung Yang, Eki Ruskartina, Budi Santosa. "Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem", Applied Soft Computing, 2017 Publication	<1%
44	fedetd.mis.nsysu.edu.tw Internet Source	<1%
45	amsdottorato.unibo.it Internet Source	<1%
46	www.unizar.es Internet Source	<1%
47	Andrew Lim, Xingwen Zhang. "A Two-Stage Heuristic with Ejection Pools and Generalized Ejection Chains for the Vehicle Routing Problem with Time Windows", INFORMS Journal on Computing, 2007 Publication	<1%
48	ojs.academypublisher.com Internet Source	<1%
49	Gong, Yue-Jiao, Jun Zhang, Ou Liu, Rui-Zhang Huang, Henry Shu-Hung Chung, and Yu-Hui Shi. "Optimizing the Vehicle Routing Problem With Time Windows: A Discrete Particle Swarm Optimization Approach", IEEE Transactions on	<1%

	Systems Man and Cybernetics Part C (Applications and Reviews), 2012. Publication	
50	Submitted to University of Nottingham Student Paper	<1%
51	Trinh, Truong Hong, and Voratas Kachitvichyanukul. "An analytical framework for the design of service delivery systems: a co-production approach", International Journal of Operational Research, 2013. Publication	<1%
52	Yonca Erdem Demirtas, Erhan Ozdemir, Umut Demirtas. "A particle swarm optimization for the dynamic vehicle routing problem", 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 Publication	<1%
53	Submitted to Brunel University Student Paper	<1%
54	Submitted to University of Macau Student Paper	<1%
55	"Handbook of Heuristics", Springer Nature America, Inc, 2018 Publication	<1%
56	Submitted to University of Kent at Canterbury	

Student Paper

<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		