Int j simul model 15 (2016) 4, 742-753
ISSN 1726-4529 Original scientific paper

VEHICLE ROUTING PROBLEM WITH TIME WINDOWS USING MULTI-OBJECTIVE CO-EVOLUTIONARY APPROACH

Wu, D. Q.[*,**,***,****,*****,******]; Dong, M.[***]; Li, H. Y.[**] & Li, F.[*******]

[*] College of Economics & Management, Shanghai Ocean University, Shanghai, 201306, China
[**] Computer Science and Technology Institute, University of South China, Hengyang, 421001, China
[***] Antai College of Economics & Management, Shanghai Jiao Tong University, Shanghai, 200240, China
[****] Artificial Intelligence Key Laboratory of Sichuan Province (Sichuan University of Science and Engineering), Zigong, 643000, China
[*****] Key Laboratory of Guangxi High Schools for Complex System & Computational Intelligence, Guangxi University for Nationalities, Nanning, 530004, China
[******] Nanjing University of Information Science and Technology, Nanjing, 210044, China
[*******] School of Computer Science and Technology, Donghua University, Shanghai, 201620, China
E-Mail: Dqw_1982@126.com, mdong@sjtu.edu.cn, Xiaoyueshen@sohu.com, lifeng@dhu.edu.cn

Abstract

This paper introduces a novel multi-objective algorithm (HMPSO) based on discrete particle swarm optimization (PSO) to solve vehicle routing problems with time windows (VRPTW). The presented HMPSO algorithm was combined with an advanced discrete PSO based on set and variable neighbourhood searches to find Pareto optimal routing solutions. These consisted of a complete routing schedule for serving the customers to minimize the two aims of travelling distance and number of vehicles. To increase the discrete PSO efficiency, a novel decoding scheme based on set was designed, and the variable neighbourhood local search was employed to explore new solutions. The experiment results were showed for a set of the Solomon's 56 VRPTW. The HMPSO algorithm was compared with some algorithms published in papers with the computational evaluations clearly supporting the high performance of the proposed HMPSO algorithm against other algorithms, and confirming that the HMPSO is an efficient algorithm because of a reasonable computational time and cost in solve VRPTW.

(Received, processed and accepted by the Chinese Representative Office.)

Key Words: Multi-Objective Optimization, Discrete Particle Swarm Optimization, Variable Neighbourhood Search, Vehicle Routing Problem with Time Windows

1. INTRODUCTION

Vehicle routing problem (VRP) is a hot topic to transportation and logistics systems. This problem designs some routes from depots to a set of geographically scattered entrepots, cities or customers etc. with a group of vehicles, which should serve customers located in distinct regions. In recent years, many vehicle routing problems in practice have been extended to vehicle routing problem with time windows.

In the last ten years, nature inspired intelligence became more and more popular [1]. Due to the advancements in evolutionary computation, in the literature, many researchers have proposed many new evolutionary algorithms and made significant efforts to solve the VRP. These include the tabu search [2], the ant system [3] and, particle swarm optimization (PSO) [4]. Ai and Kachitvichyanukul [5] put forward a novel PSO for a VRP with simultaneous pick-up and delivery. In [6] they reported some promising results for the capacitated VRP (CVRP). Moghaddam et al. [7] presented a new decoding algorithm to solve the CVRP. Marinakis and Marinaki [8] proposed a hybrid genetic-Particle Swarm Optimization Algorithm, to not only solve the VRPTW with fewer than ten customers but also to deal with the large-scale VRPTW. The PSO algorithm was tested on some benchmarks of Solomon's

DOI:10.2507/IJSIMM15(4)CO19

100-customer, but performed unsatisfactorily in datasets R [9]. Repoussis et al. [10] proposed a novel arc-guided evolution algorithm; there were parallel structures to solve the VRPTW. Alvarenga et al. [11] combined the genetic algorithm (GA) with a set partition formulation; Malmborg [12] proposed a specific crossover operator based on graph theory for the VRPTW. The ant colony (ACO) has been applied to solve capacitated vehicle routing problems successfully [13-15]. Baños et al [16] introduced a hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. To solve vehicle routing problem with simultaneous pickup and delivery, Goksal et al. [17] described a hybrid discrete particle swarm optimization.

The majority of real life problems are not limited to minimizing one object, but other objectives can also be considered. To solve such a multi-objective optimization problem, a hybrid multi-objective PSO (HMPSO) combined with a discrete particle swarm optimization algorithm was used. This was based on the set and variable neighbourhood search heuristic to find Pareto optimal routing solutions for VRPTW. The algorithm is suitable for very large-scale vehicle routing problems within a short computational time.

2. VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

The VRPTW is a complex combinatorial optimization problem which is NP-hard. In the VRPTW, a service time, the customers' demand can only be supplied once by exactly one vehicle, the customers have predefined requirements of goods, are located in different geographical regions. The each customer and depot both have time windows. The distances between customers are measured by Euclidean distances, and the total distance travelled by all the vehicles. The VRPTW can be defined on a non-directed complete graph $G(V, E)$, where vertices $V = \{0, \ldots, N\}$ correspond to the depot and the customers, and the edges $e \in E$ $\{(i,j): i,j \in V\}$ correspond to the edges between them [18].

Decision variable:

$$x_{ij}^{k} = \begin{cases} 1 & \text{the link from customer node i to j is visited by vehicle k} \\ 0 & \text{otherwise} \end{cases}$$

Minimize total travel distance (TD): $TD = \sum_{k=1}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} C_{ij} X_{ij}^{k}$ (1)

Minimize number of vehicles (NV): $NV = |K|$ (2)

Subject to:

$$X_{ij}^{k} = 0 \qquad \left(\forall i \in \{0,\ldots,N\}, \forall k \in \{0,\ldots,K\}\right) \tag{3}$$

$$X_{ij}^{k} \in \{0,1\} \qquad \left(\forall i,j \in \{0,\ldots,N\}, \forall k \in \{0,\ldots,K\}\right) \tag{4}$$

$$\sum_{k=1}^{K} \sum_{i=1}^{N} X_{ij}^{k} = 1 \qquad \left(\forall j \in \{1,\ldots,N\}\right) \tag{5}$$

$$\sum_{k=1}^{K} \sum_{j=1}^{N} X_{ij}^{k} \le K \tag{6}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} X_{ij}^{k} d_j \le Q \qquad \left(\forall k \in \{1,\ldots,K\}\right) \tag{7}$$

$$a_j \le s_{kj} \le b_j \qquad \left(\forall i,j \in \{0,\ldots,N\}, \forall k \in \{1,\ldots,K\}\right) \tag{8}$$

$$\sum_{j=1}^{N} X_{1j}^{k} - \sum_{j=1}^{N} X_{j1}^{k} = 0 \qquad \left(\forall k \in \{1,\ldots,K\}\right) \tag{9}$$

$$s_{kj} + C_{ij} - L\left(1 - X_{ij}^{k}\right) \le s_{kj} \qquad \left(\forall i,j \in \{0,\ldots,N\}, \forall k \in \{1,\ldots,K\}\right) \tag{10}$$

where: a_j is the earliest time for customer j to allow the service; b_j is the latest time for customer j to allow the service; C_{ij} is the cost for travelling from node i to node j (here, C_{ij} is considered as the distance or time required for travelling from node i to node j); d_j is the demand at customer j; Q is the loading capacity of each vehicle; N is the number of customers plus the depot (the depot is denoted with number 0, and the customers are denoted as $1, ..., N$) and K is the maximum number of vehicles that can be used.

The VRPTW can be stated as follows: Eqs. (1) and (2) are the objectives function of the problem, with TD and TV being the distance travelled by vehicles and the total number of vehicles used to serve the customers. Eq. (3) denotes that a vehicle must travel from one node to another. Eq. (4) indicates that X_{ij}^{k} is equal to 1 if vehicle k goes from node i to node j, and is equal to 0 otherwise. Eq. (5) states that a customer is visited once by exactly one vehicle. By specifying the constraint, Eq. (6) specifies that there are up to K routes going out of the delivery depot. In Eq. (7), it is consider that for a given vehicle k, the load that has to be transported to complete the route assigned to it cannot exceed its capacity Q. Eq. (8) ensures that time windows are observed. Given a large value, l, Eq. (9) guarantees that vehicles depart from and return to the depot. Let s_{kj} be the sum of the distances travelled by a vehicle k before arriving with the customer j, the inequality represented in Eq. (10) specifies that, if the vehicle k is travelling from customer i to j, the vehicle cannot arrive at customer j before $s_{kj} + s_{ij}$. As specified by El-Sherbeny [19], the variable s_{kj} corresponds s to the time the vehicle k starts to service the customer j. If the vehicle k does not service j, s_{kj} is not calculated.

3. MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION ALGORITHM FOR VRPTW

3.1 Initialize population

Particle swarm optimization (PSO) was originally designed and developed by Eberhart and Kennedy [20], which is inspired by social behaviour simulation. The original PSO was used for continuous optimizations. Chen et al. [21, 22] proposed a set-based PSO to characterize the discrete search space of combinatorial optimization problems. PSO is a population-based search algorithm that operated on the simulation of the social behaviour of birds within a flock. Each particle represents a potential solution to the problem and searches around in a multi-dimensional search space. They simulated birds' swarm behaviour according to its experience and the best experience of particle.

In this paper, the Push Forward Insertion Heuristic (PFIH) method was used in the population initialization [23, 24]. Once the first customer is selected for the current route, the other customer was selected from the set of unrouted customers, which minimizes the total insertion cost between every edge in the current route. The relation (11) defines the first customer in each new route.

$$c_i = -\alpha d_{0i} + \beta b_i + \gamma ({p_i}/{360}) d_{0i} \tag{11}$$

where: b_i is the upper time window limit to reach the customer i, α is 0.7, β is 0.1, γ is 0.2, d_{0i} is the distance from central depot to the customer i, p_i is the polar coordinate angle of the customer i from the central depot.

The space of the VRPTW problem is a directed complete graph $G = (V, E)$, where E is the arc set of the link from customer node i to j, v is the set of customer node. If they satisfy the constraints, the candidate solutions are judged to be feasible solutions, and each candidate solution spans a subgraph of G.

The search space is represented by a universal set E, the arc set of the link from customer node i to j. n dimensions were divided form the elements in E, one dimension represents a subset equal E^i $(i = 0, 1, \dots, n)$, i.e., $E = E^0 \cup E^1 \cup E^2 \cup \dots \cup E^n$.

(1) X is considered to be a candidate solution and satisfies $X \subseteq E$, i.e. $X = X^0 \cup X^1 \cup X^2 \cup \ldots \cup X^n$, $X^i \subseteq E^i$ $(i = 1, \ldots, n)$.

(2) The objective of the problem is to find a feasible solution X^* if and only if it satisfies all the constraints, X^* is identified as a feasible solution. This feasible initial population is built as depicted in Fig. 1.

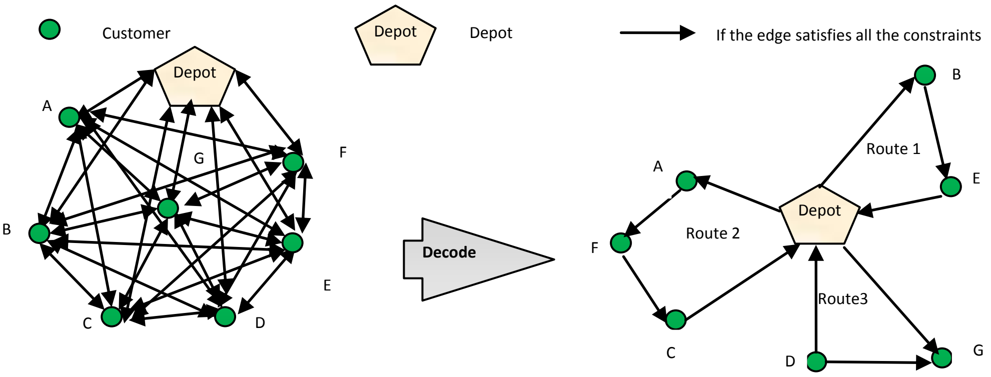
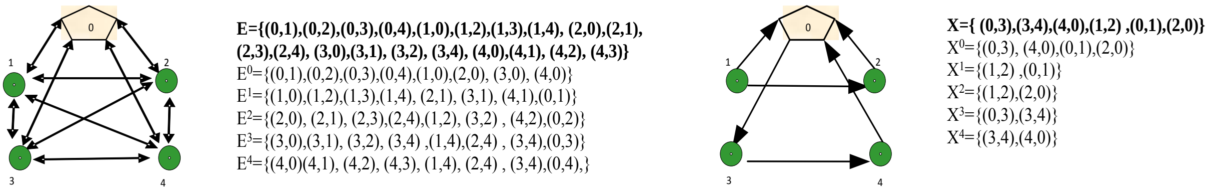


Figure 1: The feasible initial population in the HMPSO.

In the HMPSO, these arc sets and possibilities update the velocity and position rules, the arc sets and possibilities correspond to the position and velocity in the HMPSO. A crisp set of arcs which is a subset of E represent the position of each particle. A set of arcs with possibilities represent the velocity. The new individual (candidate solution) is constructed by the selected arcs. In the construction, with a larger possibility, an arc associated is more likely to be selected. In the VRPTW, considering the time constraint, every customer was served in the order. Since arc (m, k) was not equal to arc (k, m), the dimensions E^m and E^k shared the different arc (m, k). Fig. 2 gives an example of the representation scheme for the VRPTW.



E={(0,1),(0,2),(0,3),(0,4),(1,0),(1,2),(1,3),(1,4), (2,0),(2,1), (2,3),(2,4), (3,0),(3,1), (3,2), (3,4), (4,0),(4,1), (4,2), (4,3)}
E^0={(0,1),(0,2),(0,3),(0,4),(1,0),(2,0), (3,0), (4,0)}
E^1={(1,0),(1,2),(1,3),(1,4), (2,1), (3,1), (4,1),(0,1)}
E^2={(2,0), (2,1), (2,3),(2,4),(1,2), (3,2) , (4,2),(0,2)}
E^3={(3,0),(3,1), (3,2), (3,4) ,(1,4),(2,4) , (3,4),(0,3)}
E^4={(4,0)(4,1), (4,2), (4,3), (1,4), (2,4) , (3,4),(0,4),}

X={ (0,3),(3,4),(4,0),(1,2) ,(0,1),(2,0)}
X^0={(0,3), (4,0),(0,1),(2,0)}
X^1={(1,2) ,(0,1)}
X^2={(1,2),(2,0)}
X^3={(0,3),(3,4)}
X^4={(3,4),(4,0)}

Figure 2: The VRPTW representation scheme.

3.2 Defined the position and the velocity

HMPSO use a particle swarm including n individuals (candidate solutions) to search Pareto optimal solutions with the position of a particle are defined by (12)-(13):

$$X_i = [X_i^0, X_i^1, \ldots, X_i^n] \tag{12}$$

$$X_i^t = \left[(m,t),(t,k)\right], m,k \in \{0,1,\ldots,t-1,t+1,n\}, m \neq k \tag{13}$$

where x_i is an arc set with each dimension X_i^t and X_i^t is composed of two arcs adjacent to customer (node) t. In (12): n is the number of nodes; t is the current node; k is the subsequent node of node t and; m is the previous node of node t. Each individual represents a route which a vehicle could take.

A set with possibilities defined the velocity of a particle:

$$V_i = [V_i^0, V_i^1, \ldots, V_i^d] \tag{14}$$

$$V_i^d = \left\{ e / p(e) \middle| (e) \in E^d \right\} \tag{15}$$

Each element $e \in E$ has a possibility $p(e) \in [0, 1]$, in the d^{th} dimension, $e \in E^d$ with possibilities $p(e)$ in v, e is an arc which link two customers, and here $p(e)$ is proportion set of the distance between nodes and total distance.

3.3 Velocity and position updating

Inspired by the work in [21], the velocity updating is shown in (16)-(19) as follows:

$$c \times V_i^d = \left\{(e)/p'(e) \middle| (e) \in E^d\right\} \qquad p'(e) = \begin{cases} 1, & if \;\; c \times p(e) > 1 \\ c \times p(e), & otherwise \end{cases} \tag{16}$$

$$V_i^d + V_j^d = \left\{(e)/\max(p_i(e), p_j(e)) \middle| (e) \in E^d\right\} \tag{17}$$

$$X_i^d - X_j^d = T^d \left\{(e) \middle| (e) \in X_i^d and \middle| (e) \notin X_j^d\right\} \tag{18}$$

$$c \times T^d = \left\{(e)/p'(e) \middle| (e) \in E^d\right\} \qquad p'(e) = \begin{cases} 1, & if \;\; e \in T^d \;\; and \;\; c > 1 \\ c, & if \;\; e \in T^d \;\; and \;\; 1 \ge c \ge 0 \\ 0, & if \;\; e \notin T^d \end{cases} \tag{19}$$

where changing the $p(e)$ value defines the operators of 'c×velocity' and 'velocity+velocity'. The subtraction of the two arc sets define operator of position-position, converting a crisp set into a set with $p(e)$ define the operator of 'c×(position-position)'.

In the position updating process, the first velocity V_i is converted into a temp set by (20).

$$Temp(V_i^d) = \left\{(e) \middle| (e)/p(e) \in V_i^d and \;\; p(e) \ge P(rand)\right\} \tag{20}$$

where $P(rand)$ is a random number between 0 and 1, for an arc(e), if and only if the velocity's corresponding possibility $p(e)$ is equal or greater than the $P(rand)$. At first X_i' is an empty set.

3.4 Variable neighbourhood search (VNS)

Variable neighbourhood search [25] is one of the very well-known local search methods. In this paper, the past best position and global best position is provided by the members in the external repository, and it constrains the particles search ability. In the proposed method, the variable neighbourhood search applies to the routes, in order to maintain the solution diversity and produce various solutions. The procedure of the variable neighbourhood search was depicted as in Fig. 3.

```
Algorithm 1: The variable neighbourhood search

For Ll(s). Step 1: Choose randomly a route $R$.
           Step 2: Choose randomly two customs $N_1$ and $N_2$ on route $R$ from external archive.
           Step 3: Swap customs $N_1$ and $N_2$ on the premise such that the insertion satisfies all
                   constraints of the VRPTW.
For L2(s). Step 1: Choose randomly two routes $R_1$ and $R_2$ from external archive.
           Step 2: Choose randomly custom $N_1$ on $R_1$ and a vehicle $N_2$ on route $R_2$ from external
                   archive.
           Step 3: Swap custom $N_1$ and $N_2$ on the premise such that the insertion satisfies all
                   constraints of the VRPTW.
For L3(s). Step 1: From external archive, select the route $R$ which has the fewest loading capacity.
           Step 2: If the insertion satisfies all constraints, insert the route into the other vehicle
                   routes.
           Step 3: In a route, if all the nodes can be inserted into other routes, then delete the route.
```

Figure 3: Pseudo code of local searches.

3.5 Redefinition of the past best position (*pbest*) updating operation

In MO problems, due to the particle searching behaviour, the current position of a particle usually is the best position. There is a useless guiding situation, in order to solve the problem and ensure the diversity of population [26]. In the proposed HMPSO algorithm, *m* dimensions of each individual were randomly chosen to learn from a random non-dominated solution, which came from external archive.

3.6 The framework of hybrid HMPSO algorithm

As we can see, the routes for a fleet of vehicles define the VRPTW. The goal of HMPSO is to find Pareto solutions for two objectives: one is the number of vehicle (*NV*), and another one is the total travel distance (*TD*) while all the constraints are satisfied as defined by Eqs. (3) to (10). In the HMPSO algorithm, a set-based representation scheme is applied first which redefines the "position" and "velocity" terms and all related operators for the discrete space, then employs a variable neighbourhood search to improve upon solutions. The procedure of the algorithm is shown in Fig. 4.

```
The HMPSO algorithm for VRPTW
Initialization population, associated velocities, Max_gen and parameter values. Evaluate all particles
and assign past best particle (pbest), global best particle (gbest), the external archive (EA).
While stopping criterion is not met
  For one Particle
    Step 1: Update position and velocity.
    Step 2: Updates the external archive using dominate concepts.
    Step 3: Execute variable neighbourhood search mechanism in the external archive and the
            updates the external archive using dominate concepts.
    Step 4: pbest updating operation and update gbest to use the members in the external archive.
  End For
End While
```

Figure 4: The Pseudo-code of HMPSO for VRPTW.

4. EXPERIMENT SIMULATION AND ANALYSIS

4.1 Test problems and parameter settings

This experiment run on a PC with a 3.8 GHz CPU and 8 GB memory, the HMPSO was coded in MATLAB R2012b. The experimental results used the standard Solomon's VRPTW benchmark problem instances available at [27]. The parameters were set in HMPSO as follows, a population size is 100, the acceleration constants $c_1 = c_2 = 1.49$, the generation number set 1000, the minimum inertia weight $w_{min} = 0.2$, the maximum inertia weight $w_{max} = 0.9$, and an external archive size is 40.

4.2 Simulation results and comparisons

In multi-objective algorithms, since the objectives to be optimized had different scales, the convergence and the diversity are very important [28], the set coverage and the hyper-volume was used to evaluate the quality of the sets of non-dominated solutions, which were proposed by Zitzler and Thiele [29] and others [30].

HMPSO suggests that the conflicting behaviour for problem C101, where the number of vehicles was increased step by step from 10 to 24 vehicles, was that the routing cost decreased from 2588.1 to 828.78 steadily. Fig. 6 shows the performance of HMPSO. Moreover, in most instances, the single Pareto was generated by HMPSO in Table I.

Compared with the best known solutions, which are reported in other papers, the average Euclidian distance measures the distance costs. Table I presents a summary of the results. The results were generated in 10 runs. In Table I the suggested approach was divided into two columns representing the *TD* and *NV*. Numbers in bold in Table I show the solution obtained by the HMPSO.

Table I: Testing result on benchmark Solomon's 56 VRPTW 100-customer instances.

Data set	Best-known result			This work (HMPSO)	
	NV	*TD*	Source	Non-dominated solutions	
				NV	*TD*
C101	10	828.94	Ombuki et al. (2006) [32]	10	828.78
C102	10	828.94	Ombuki et al. (2006) [32]	10	827.32
C103	10	828.06	Ombuki et al. (2006) [32]	10	826.06
C104	10	824.78	Ombuki et al. (2006) [32]	10	822.99
C105	10	828.94	Ombuki et al. (2006) [32]	10	828.94
C106	10	828.94	Ombuki et al. (2006) [32]	10	826.82
C107	10	828.94	Ombuki et al. (2006) [32]	10	826.82
C108	10	828.94	Ombuki et al. (2006) [32]	10	826.82
C109	10	828.94	Yu et al. (2011) [33]	10	826.82
C201	3	591.58	Yu et al. (2011) [33]	3	589.58
C202	3	591.56	Yu et al. (2011) [33]	3	589.56
C203	3	593.25	Li and Lim (2003) [37]	3	591.17
C204	3	595.55	Yu et al. (2011) [33]	3	590.6
C205	3	588.88	Yu et al. (2011) [33]	3	588.88
C206	3	588.49	Yu et al. (2011) [33]	3	588.49
C207	3	588.88	Yu et al. (2011) [33]	3	588.49
C208	3	588.03	Tan et al. (2006) [31]	3	588.03
R101	19	1677	Ghoseiri and Ghan. (2010) [38]	19	1655.23
R102	18	1491.18	Ghoseiri and Ghan. (2010) [38]	17	1453
R103	13	1175.67	Tan et al. (2006) [31]	14	1243.22
R104	10	974.2	Tan et al. (2006) [31]	10	979.21
R105	15	1346.12	Kallehauge et al. (2006) [34]	16	1354.72
R106	13	1265.36	Yu et al. (2011) [33]	13	1234.6
R107	11	1051.84	Kallehauge et al. (2006) [34]	11	1101.25
R108	10	954.03	Tan et al. (2006) [31]	9	960.78
R109	12	1101.99	Yu et al. (2011) [33]	12	1032.23
R110	11	1112.21	Ombuki et al. (2006) [32]	11	1110.64
R111	10	1096.72	Ombuki et al. (2006) [32]	10	1045.29
R112	10	976.99	Ombuki et al. (2006) [32]	10	953.26
R201	7	1214.22	Yu et al. (2011) [33]	8	1199.75
R202	5	1105.2	Yu et al. (2011) [33]	6	1077.66
R203	4	960.14	Yu et al. (2011) [33]	5	938.48
R204	3	789.72	Tan et al. (2006) [31]	3	784.53
R205	3	994.42	Ombuki et al. (2006) [32]	3	954.68
R206	4	1050.26	Yu et al. (2011) [33]	3	1104
R207	4	870.33	Yu et al. (2011) [33]	3	814.78
R208	3	777.72	Yu et al. (2011) [33]	2	731.24
R209	3	934.21	Yu et al. (2011) [33]	3	855
R210	3	954.12	Bent and Van Hent. (2004) [35]	5	949.24
R211	2	892.71	Berger and Barkaoui (2004) [36]	4	886.46
RC101	14	1650.14	Yu et al. (2011) [33]	15	1623.53
RC102	13	1470.26	Tan et al. (2006) [31]	13	1497.4

Table I (continuation)

RC103	11	1277.11	Yu et al. (2011) [33]	12	1196.34
RC104	10	1159.37	Yu et al. (2011) [33]	10	1135.48
RC105	16	1590.25	Ombuki et al. (2006) [32]	15	1627.43
RC106	13	1371.69	Tan et al. (2006) [31]	13	1383.65
RC107	11	1222.16	Tan et al. (2006) [31]	11	1291.6
RC108	11	1133.9	Tan et al. (2006) [31]	11	1142.73
RC201	6	1134.91	Tan et al. (2006) [31]	5	1301.45
RC202	5	1130.53	Tan et al. (2006) [31]	5	1164.98
RC203	4	1026.61	Tan et al. (2006) [31]	6	1054.22
RC204	3	799.12	Tan et al. (2006) [31]	4	789.43
RC205	5	1295.46	Tan et al. (2006) [31]	5	1327.82
RC206	4	1139.55	Ombuki et al. (2006) [32]	5	1119.5
RC207	4	1040.67	Tan et al. (2006) [31]	4	1083.4
RC208	3	829.69	Ombuki et al. (2006) [32]	3	857.43

In each category, based on the average number of vehicles cost of routing, we compare the routing performance between five popular algorithms and HMPSO. Table II shows the result. In category C1, the number pair (10, 826.86) means that it exceeds the 9 instances in C1, 10 vehicles were deployed, and the total travelling distance was 826.86. The result shows that, for category C1, HMPSO got smallest cost in *TD* and *NV*. For the categories of C2, R1, RC1 and RC2, the smallest routing costs also were produced. For category R2, compared to the algorithm in paper [31], the average number of vehicles was 4.2 % higher than with the HMPSO. The results indicated that HMPSO performed equally well for two objects, which were optimized at the same time.

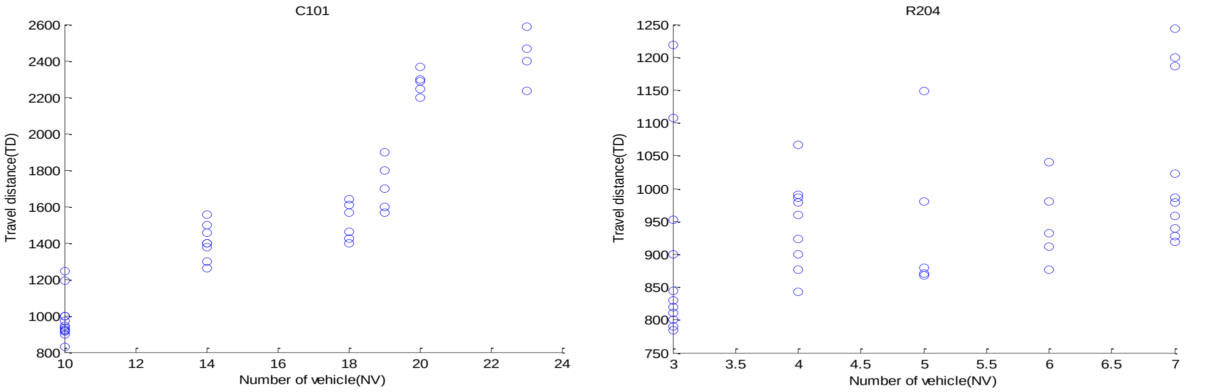


Figure 5: Conflicting behaviour for problem C101 and R204.

Table II: Comparison among different heuristics.

Data set	Best-known	PHGA [36]	HMOEA [31]	ACO-N [33]	MGPGA [38]	THLS [35]	HMPSO
C1	*NV*	10	10	10	10	10	10
	TD	828.38	832.13	841.92	828.45	827.00	826.86
C2	*NV*	3	3	3.3	3	3	3
	TD	589.86	589.86	612.75	591.48	590.00	589.35
R1	*NV*	12	12.16	13.1	13.6	12.92	12.7
	TD	1217.73	1211.55	1213.16	1247.75	1187.00	1176.95
R2	*NV*	2.73	3	4.6	3.76	3.51	4.09
	TD	967.75	1001.12	952.3	1044.88	951.00	911.34
RC1	*NV*	11.63	12.25	12.7	13.24	12.74	12.5
	TD	1382.42	1418.77	1415.62	1414.81	1355.0	1362.27
RC2	*NV*	3.25	3.37	5.6	3.95	4.25	4.75
	TD	1129.19	1170.93	1120.37	1193.06	1067.00	1094.65

Computing the hyper-volume metric *HV* required an appropriate reference point z to be set. Since every instance had an obvious maximum solution, with longest travel distance *TD* and largest number of vehicles *NV*. For every instance, the reference point was set at $Z=(NV:TD)$. From the 10 runs performed, there was a set of 10 *HV* values, starting from the same initial population, the non-dominated sets achieved by HMPSO, HMOEA, ACO-N and MGPGA. Table III shows normalized values of the hyper-volume metric over the other algorithms. For categories C1, C2 and R1, between the three algorithms, there are small differences. For categories R2, RC1, RC2, the HMPSO exhibited significant improvement over the other three algorithms. On average, 39.09 %, 30.9 % and 34.5 % of the non-dominated solutions obtained by HMOEA, ACO-N and MGPGA respectively were dominated by solutions of HMPSO. So the results validate the good performance of HMPSO.

Table III: Median hyper-volume obtained by all the methods.

Data set	Type C1	Type C2	Type R1	Type R2	Type RC1	Type RC2	AVG
HMOEA [31]	0.079	0.067	0.070	0.083	0.076	0.084	0.0765
ACO-N [33]	0.080	0.070	0.072	0.053	0.081	0.076	0.072
MGPGA [35]	0.078	0.069	0.064	0.081	0.073	0.081	0.074
HMPSO	0.077	0.064	0.062	0.044	0.033	0.054	0.055

The result of set coverage obtained by all of the methods obtained is shown in Table IV. The quality of the solutions got by HMPSO (SC metric of 0.6) was better than the HMOEA (SC metric of 0.49), ACO-N (SC metric of 0.52) and MGPGA (SC metric of 0.51) in the R1 problem. The solution obtained by HMPSO was considerably better at the C1 problem. In RC1 and RC2, the quality of the solutions obtained by HMPSO was significantly better than other algorithms.

Table IV: Set coverage obtained by all the methods.

Data set	HMOEA [31]			ACO-N [33]			MGPGA [38]			HMPSO		
Algorithm	Type R1	Type R2	Type C1	Type R1	Type R2	Type C1	Type R1	Type R2	Type C1	Type R1	Type R2	Type C1
HMOEA				0.51	0.27	0.53	0.46	0.50	0.62	0.81	0.59	0.51
ACO-N	0.56	0.73	0.49				0.54	0.70	0.56	0.52	0.44	0.50
MGPGA	0.64	0.52	0.38	0.58	0.32	0.44				0.48	0.55	0.71
HMPSO	0.27	0.43	0.49	0.48	0.64	0.57	0.52	0.45	0.29			
AVG	0.49	0.56	0.45	0.52	0.41	0.51	0.51	0.55	0.49	0.60	0.53	0.57
Data set	HMOEA [31]			ACO-N [33]			MGPGA [38]			HMPSO		
Algorithm	Type C2	Type RC1	Type RC2	Type C2	Type RC1	Type RC2	Type C2	Type RC1	Type RC2	Type C2	Type RC1	Type RC2
HMOEA				0.41	0.43	0.73	0.53	0.30	0.51	0.60	0.73	0.55
ACO-N	0.66	0.57	0.23				0.26	0.59	0.44	0.64	0.73	0.84
MGPGA	0.49	0.70	0.42	0.74	0.44	0.59				0.45	0.76	0.69
HMPSO	0.42	0.39	0.52	0.40	0.27	0.16	0.63	0.24	0.31			
AVG	0.52	0.55	0.39	0.52	0.38	0.49	0.47	0.38	0.42	0.56	0.74	0.69

With consider the Deb’s spacing metric, Table V summarizes the results got by all of these algorithms. The average of the non-dominated solutions got by HMPSO was 0.28, which is also better than those obtained by other algorithms. The non-dominated solutions obtained by HMPSO had a more concentrated distribution in most problem instances, compared to other algorithms. From above, the experimental results show that HMPSO outperformed the other peer algorithms in terms of both the convergence speed and solution accuracy. The reasons for this was that the HMPSO combined the novel discrete particle swarm optimization and the variable neighbourhood search, meaning that it activated the particles search process and decreased the convergence speed of PSO. The particles moved into their local optimum by a faster and efficient way, the variable neighbourhood search provides useful information guiding particles to find better solutions.

Table V: The results for Deb's spacing metric.

Data set	Type R1	Type R2	Type C1	Type C2	Type RC1	Type RC1	AVG
HMOEA[31]	0.84	0.73	0.87	0.85	0.67	0.69	0.78
ACO-N [33]	0.67	0.57	0.79	0.72	0.53	0.51	0.63
MGPGA [38]	0.54	0.48	0.6	0.43	0.27	0.32	0.44
HMPSO	0.28	0.32	0.39	0.27	0.16	0.26	0.28

5. CONCLUSION

In this paper, a hybrid multi-objective PSO (HMPSO) was proposed to solve the VRPTW. In HMPSO, in order to maintain the diversity of PSO and explore new solutions, the HMPSO employed variable neighbourhood search to improve upon solutions when PSO was fall to local optima. Solomon's 56 VRPTW was used to validate that HMPSO algorithm is an effective tool for VRPTW. Moreover, the results show the robustness of the algorithm.

ACKNOWLEDGEMENT

The authors would like to thank all the reviewers for their constructive comments. This research was supported by the project of Ministry of Education of Hunan province (No. 2016NK2135), the project of Hunan Provincial Department of Science and Technology (No. 2015JC3089), the open fund of Key Laboratory of Guangxi High Schools for Complex System & Computational Intelligence (No. 15CI07Y), the open project program of artificial intelligence key laboratory of Sichuan province (No. 2015RYJ01), the China Statistical Science Research Project (No. 2015LZ17), the Natural Science Foundation of Shanghai, China (No. 16ZR1401100), the Shanghai Academy of Productivity young scholars' research assistant fund, the social science project in Hunan province.

REFERENCES

[1] Sariklis, D.; Powell, S. (2000). A heuristic method for the open vehicle routing problem, *Journal of the Operational Research Society*, Vol. 51, No. 5, 564-573, doi:10.1057/palgrave.jors.2600924

[2] Fu, Z.; Eglese, R.; Li, L. Y. O. (2005). A new tabu search heuristic for the open vehicle routing problem, *Journal of the Operational Research Society*, Vol. 56, No. 3, 267-274, doi:10.1057/palgrave.jors.2601817

[3] Li, X.; Tian, P. (2006). An ant colony system for the open vehicle routing problem, *ANTS'06 Proceedings of the 5th International Conference on Ant Colony Optimization and Swarm Intelligence*, 356-363

[4] Wu, D.; Zheng, J. (2012). A dynamic multistage hybrid swarm intelligence optimization algorithm for function optimization, *Discrete Dynamics in Nature and Society*, Vol. 2012, Paper 578064, 22 pages, doi:10.1155/2012/578064

[5] Ai, T. J.; Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery, *Computers & Operations Research*, Vol. 36, No. 5, 1693-1702, doi:10.1016/j.cor.2008.04.003

[6] Ai, T. J.; Kachitvichyanukul, V. (2009). Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem, *Computers & Industrial Engineering*, Vol. 56, No. 1, 380-387, doi:10.1016/j.cie.2008.06.012

[7] Moghaddam, B. F.; Ruiz, R.; Sadjadi, S. J. (2012). Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm, *Computers & Industrial Engineering*, Vol. 62, No. 1, 306-317, doi:10.1016/j.cie.2011.10.001

[8] Marinakis, Y.; Marinaki, M. (2010). A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem, *Expert Systems with Applications*, Vol. 37, No. 2, 1446-1455, doi:10.1016/j.eswa.2009.06.085

[9] Lin, C.-T. (2008). Using predicting particle swarm optimization to solve the vehicle routing problem with time windows, *2008 IEEE International Conference on Industrial Engineering and Engineering Management Proceedings*, 810-814, doi:10.1109/IEEM.2008.4737982

[10] Repoussis, P. P.; Tarantilis, C. D.; Ioannou, G. (2009). Arc-guided evolutionary algorithm for the vehicle routing problem with time windows, *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 3, 624-647, doi:10.1109/TEVC.2008.2011740

[11] Alvarenga, G. B.; Mateus, G. R.; de Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows, *Computers & Operations Research*, Vol. 34, No. 6, 1561-1584, doi:10.1016/j.cor.2005.07.025

[12] Malmborg, C. J. (1996). A genetic algorithm for service level based vehicle scheduling, *European Journal of Operational Research*, Vol. 93, No. 1, 121-134, doi:10.1016/0377-2217(95)00185-9

[13] Ventresca, M.; Ombuki-Berman, B.; Runka, A. (2013). Predicting Genetic Algorithm Performance on the Vehicle Routing Problem Using Information Theoretic Landscape Measures, Middendorf, M.; Blum, C. (Eds.), *Evolutionary Computation in Combinatorial Optimization*, Springer-Verlag, Berlin, 214-225, doi:10.1007/978-3-642-37198-1_19

[14] Fu, Z.-J.; Sun, X.-M.; Liu, Q.; Zhou, L.; Shu, J.-G. (2015). Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing, *IEICE Transactions on Communications*, Vol. E98-B, No. 1, 190-200, doi:10.1587/transcom.E98.B.190

[15] Xie, S.; Wang, Y. (2014). Construction of tree network with limited delivery latency in homogeneous wireless sensor networks, *Wireless Personal Communications*, Vol. 78, No. 1, 231-246, doi:10.1007/s11277-014-1748-5

[16] Baños, R.; Ortega, J.; Gil, C.; Marquez, A. L.; de Toro, F. (2013). A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows, *Computers & Industrial Engineering*, Vol. 65, No. 2, 286-296, doi:10.1016/j.cie.2013.01.007

[17] Goksal, F. P.; Karaoglan, I.; Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery, *Computers & Industrial Engineering*, Vol. 65, No. 1, 39-53, doi:10.1016/j.cie.2012.01.005

[18] Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows, *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 5, 668-679, doi:10.1016/j.trc.2009.08.005

[19] El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods, *Journal of King Saud University (Science)*, Vol. 22, No. 3, 123-131, doi:10.1016/j.jksus.2010.03.002

[20] Kennedy, J.; Eberhart, R. (1995). Particle swarm optimization, *IEEE International Conference on Neural Networks – Conference Proceedings*, 1942-1948, doi:10.1109/ICNN.1995.488968

[21] Chen, A.-L., Yang, G.-K.; Wu, Z.-M. (2006). Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem, *Journal of Zhejiang University – Science A*, Vol. 7, No. 4, 607-614, doi:10.1631/jzus.2006.A0607

[22] Chen, W.-N.; Zhang, J.; Chung, H. S. H.; Zhong, W.-L.; Wu, W.-G.; Shi, Y.-H. (2010). A novel set-based particle swarm optimization method for discrete optimization problems, *IEEE Transactions on Evolutionary Computation*, Vol. 14, No. 2, 278-300, doi:10.1109/TEVC.2009.2030331

[23] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, Vol. 35, No. 2, 254-265, doi:10.1287/opre.35.2.254

[24] Ren, Y.-J.; Shen, J.; Wang, J.; Han, J.; Lee, S.-Y. (2015). Mutual verifiable provable data auditing in public cloud storage, *Journal of Internet Technology*, Vol. 16, No. 2, 317-323, doi:10.6138/JIT.2015.16.2.20140918

[25] Ishibuchi, H.; Yoshida, T.; Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multi-objective permutation flow shop scheduling, *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 2, 204-223, doi:10.1109/TEVC.2003.810752

[26] Liang, J. J.; Qin, A. K.; Suganthan, P. N.; Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, 281-295, doi:10.1109/TEVC.2005.857610

[27] Solomon, M. M. VRPTW Benchmark Problems / Best Known Solutions, from *http://w.cba.neu.edu/~msolomon/problems.htm*, accessed on 05-05-2016

[28] Coello, C. A.; Lamont, G. B.; Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (Genetic and evolutionary computation series), Springer, New York

[29] Zitzler, E.; Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, 257-271, doi:10.1109/4235.797969

[30] Garcia-Najera, A.; Bullinaria, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows, *Computers & Operations Research*, Vol. 38, No. 1, 287-300, doi:10.1016/j.cor.2010.05.004

[31] Tan, K. C.; Chew, Y. H.; Lee, L. H. (2006). A hybrid multi-objective evolutionary algorithm for solving vehicle routing problem with time windows, *Computational Optimization and Applications*, Vol. 34, No. 1, 115-151, doi:10.1007/s10589-005-3070-3

[32] Ombuki, B.; Ross, B. J.; Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows, *Applied Intelligence*, Vol. 24, No. 1, 17-30, doi:10.1007/s10489-006-6926-z

[33] Yu, B.; Yang, Z. Z.; Yao, B. Z. (2011). A hybrid algorithm for vehicle routing problem with time windows, *Expert Systems with Applications*, Vol. 38, No. 1, 435-441, doi:10.1016/j.eswa.2010.06.082

[34] Kallehauge, B.; Larsen, J.; Madsen, O. B. G. (2006). Lagrangian duality applied on vehicle routing problem with time windows, *Computers & Operations Research*, Vol. 33, No. 5, 1464-1487, doi:10.1016/j.cor.2004.11.002

[35] Bent, R.; Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows, *Transportation Science*, Vol. 38, No. 4, 515-530, doi:10.1287/trsc.1030.0049

[36] Berger, J.; Barkaoui, M. (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows, *Computers & Operations Research*, Vol. 31, No. 12, 2037-2053, doi:10.1016/S0305-0548(03)00163-1

[37] Li, H.-B.; Lim, A. (2003). Local search with annealing-like restarts to solve the VRPTW, *European Journal of Operational Research*, Vol. 150, No. 1, 115-127, doi:10.1016/S0377-2217(02)00486-1

[38] Ghoseiri, K.; Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm, *Applied Soft Computing*, Vol. 10, No. 4, 1096-1107, doi:10.1016/j.asoc.2010.04.001