Vehicle Routing Problem with Time Windows: A Hybrid Particle Swarm Optimization Approach

Xiaoxiang Liu, Weigang Jiang, Jianwen Xie
Department of Computer Science of Zhuhai College,
Jinan University, Zhuhai, Guangdong, 519070, China
tlxx@jnu.edu.cn, jfkdr@163.com, kenxie.4@163.com

Abstract

Vehicle routing problem (VRP) is a well-known combinatorial optimization and nonlinear programming problem seeking to service a number of customers with a fleet of vehicles. This paper proposes a hybrid particle swarm optimization (HPSO) algorithm for VRP. The proposed algorithm utilizes the crossover operation that originally appears in genetic algorithm (GA) to make its manipulation more readily and avoid being trapped in local optimum, and simultaneously for improving the convergence speed of the algorithm, level set theory is also added to it. We employ the HPSO algorithm to an example of VRP, and compare its result with those generated by PSO, GA, and parallel PSO algorithms. The experimental comparison results indicate that the performance of HPSO algorithm is superior to others, and it will become an effective approach for solving discrete combinatory problems.

1. Introduction

Vehicle Routing Problems (VRP) was proposed for the first time by Dantzig and Ramser [1] in 1959. This problem is a complex and challenging combinatorial optimization task, which consists in designing the optimal set of routes for fleet of vehicles in order to serve a given set of customers. The interest in VRP is motivated by its practical relevance as well as by its considerable difficulty. Theoretically, VRP is a NP-hard problem, which means that it is believed that one may never find a computational technique that will guarantee optimal solutions to larger instances for such problems [2].

Vehicle Routing Problem with Time Window (VRPTW) is an extension of normal VRP, encountered very frequently in making decision about the distribution of goods and services. The problem involves a fleet of vehicles set-off from a depot to serve a number of customers, at diverse geographic locations, with various demands and within specific time windows before returning to the depot eventually. Because a great deal of problem can be transformed into the VRPTW problem to deal with (e.g. post mailing, scheduling of trains and buses), and the qualities of their solutions directly affect the qualities of service, the research of VRPTW bas been paid more and more attention.

The method of solving VRPTW can be broadly classified into three generations [3]: the first generation was simple heuristic method developed in the 60's and 70's, which were mainly based on local search; the second generation was mathematical programming based heuristic, which were near-optimization algorithms that are very different from normal heuristics; the third generation is optimization algorithm and artificial intelligence methods, such as genetic algorithm, Tabu search, simulated annealing, improved PSO algorithm, etc.

PSO (Particle Swarm Optimization) [4] is a bionic algorithm which simulates the flight of flocks of birds, making good effect for the various multidimensional continuous space optimization problems, but less in research and application of discrete field [5]. In PSO algorithm, very few parameters are needed to be adjusted, which makes it particularly easy to implement. At present, there are lots of best-known improved PSO algorithms, such as PSOBC [6] algorithm, DRPSO algorithm [7] and so on. This paper proposes a HPSO, which is based on genetic algorithm and level set, to solve VRPTW. The results of the experiment show desirable effect.

2. Description and mathematical model for VRPTW

VRPTW is generally described as: given a central warehouse, owned K vehicles, the capacity of each vehicle is denoted q_k (k=1,2,…,K); now, transport tasks of L delivery points should be accomplished, denoted by 1,2,…,L. The cargo of the i th delivery point is g_i (i=1,2,…,L). The time needed to complete the tasks (loading or unloading) of the i th delivery point is t_i. And the task i must be completed in time window [ET_i, LT_i], in which ET_i denotes the earliest start time for the task i and LT_i denotes the latest start time for the task i. If vehicles arrive delivery point i earlier than ET_i, waiting occurs; if later than LT_i, the task i will be delayed. We should get a solution for vehicle running route with transport demands and minimal travel cost.

VRPTW can be divided into two categories: hard time widow VRPTW and soft time window VRPTW. Hard time window VRP is that each task must be completed within the specific time. However, soft time window VRP is that if a task can not be completed within the specific time, given a punishment.

In this paper, we use the mathematical model presented by literature [8], giving 0 to the central warehouse and giving 1,2,…,L to delivery points. Both tasks and central warehouse are denoted by node i (i=0, 1,…, L). Variables are defined as follows:

$$y_{ki} = \begin{cases} 1 & \text{the task of delivery point } i \text{ is} \\ & \text{completed by vehicle } k \\ 0 & \text{otherwise} \end{cases},$$

$$x_{ijk} = \begin{cases} 1 & \text{vehicle } k \text{ travels from point } i \\ & \text{to point } j \\ 0 & \text{otherwise} \end{cases}.$$

c_{ij} is the cost of transportation from the point i to the point j, it could be the meanings of distance, cost, time and so on. s_i is the time when vehicles arrive at the task i. P_E is the unit time of waiting cost when vehicle arrive at task i before ET_i. P_L is the unit cost of delay fines when vehicle arrive at task i after LT_i. If vehicles arrive at task point i before ET_i, opportunity cost $P_E(s_i\text{-}ET_i)$ will be increased; otherwise, if arriving after LT_i, punishment cost $P_L(LT_i\text{-}s_i)$ will be increased.

Mathematical model of vehicle optimizing scheduling is as follows:

$$\min z = \sum_i \sum_j \sum_k c_{ij} x_{ijk} + p_E \sum_{i=1}^{L} \max(ET_i - s_i, 0) + p_L \sum_{i=1}^{L} \max(s_i - ET_i, 0) \quad (1)$$

$$\begin{cases} s.t. \ \sum_i g_i y_{ki} \le q_k \ \ \forall k & (2) \\ \sum_k y_{ki} = 1 \quad i = 1,2,\cdots,L & (3) \\ \sum_i x_{ijk} = y_{kj} \quad j = 0,1,\cdots,L;\ \forall k & (4) \\ \sum_j x_{ijk} = y_{ki} \quad i = 0,1,\cdots,L;\ \forall k & (5) \\ x_{ijk}, y_{ki} = 0 \text{ or } 1 \quad i,j = 0,1,\cdots,L;\ \forall k & (6) \end{cases}$$

This model is widely used. It can be converted into a mathematical model of different issues by different setting of the parameters. When Eq. (1) $P_E=P_L\to\infty$, the above model is the hard time window VRP. If Eq. (1) ET_i=0, $LT_i\to\infty$, VRPTW model become a common VRP. If only one car is used, the problem will become TSP problem. If removing constraint (2), it becomes TSPTW problem.

3. Basic particle swarm algorithm

PSO was presented by Kennedy and Eberhart [6] in 1995. The algorithm simulates flight behavior of flocks of birds. Birds aim to achieve optimization results through a collective collaboration between the groups. In the PSO system, each candidate solution is called a "particle". A number of particles coexist and cooperate to find optimization. Each particle "flies" to a better position in problem space in accordance with its own "experience" and the best "experience" of the adjacent particle swarm, searching the optimal solution.

Mathematical notation of PSO algorithm is defined as follows: Assume searching space is D-dimensional and the total number of particles is n. The ith particle location is denoted by the vector: $X_i=(x_{i1}, x_{i2},\ldots, x_{iD})$; The past optimal location of the ith particle in the "flight" history (that is, the location corresponds optimal solution) is $P_i=(p_{i1}, p_{i2},\ldots, p_{iD})$. The past optimal location P_g of the gth particle is optimal in all of P_i(i=1,2,…,n); The location changing rate (speed) of the ith particle is denoted by the vector $V_i=(v_{i1},v_{i2},\ldots,v_{iD})$. The location of each particle changes by the following formula:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand() \times [p_{id}(t) - x_{id}(t)] + c_2 \times rand() \times [p_{gd}(t) - x_{id}(t)] \quad (7)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (1 \le i \le n;\ 1 \le \mathrm{d} \le D) \quad (8)$$

where c_1, c_2 are positive constants called accelerating factor; *rand*() is a random number between 0 and 1; w is called inertia factor; w, set a litter greater, is suited to a wide range of exploration to solution space while smaller is suited to a small range. $[X_{min}, X_{max}]$ is the changing range of particle location. $[v_{min}, v_{max}]$ is the changing range of speed. If the location and speed exceed boundary range in iteration, given boundary value. Based on analysis of the above parameters, Maurice Clerc provides the parameter conditioning of PSO algorithm convergence [9].

4. A hybrid particle swarm optimization algorithm to solve VRPTW

In the basic PSO algorithm, the speed of particles v is restricted by a maximum speed v_{max}. It decides searching precision in the solution space. If v_{max} is too big, particles may miss the optimal solution; otherwise, the particles may get into a local search space and can not carry through a globe search [10]. Therefore, in this paper, the idea of crossover operation of genetic algorithm is brought into PSO algorithm. Meanwhile, because PSO has the problem of low-speed convergence, level set theory is also employed for solving this deficiency. In a word, this paper applies both technique of GA and level set theory to improve standard PSO algorithm, thus proposing a combined PSO algorithm that is suitable to VRPTW.

4.1. Particle coding

This paper makes reference to the encoding manner of literature [11]. For the VRPTW problem with L customer points and K vehicles, each particle has L+K+1 dimensional vector, in which the order of each element value denotes the delivery order of the customer point in the total path.

For example: Assume the number of customers is 8 and the number of vehicles is 3. If the location vector of the particle is:

Warehouse No: 1 2 3 4 5 6 7 8 0 0

X=[6 8 4 1 2 9 5 10 3 7]

Then the total path of corresponding solution of the particle is: 0→4→5→0→3→7→1→0→2→6→8→0, corresponding separate path is:

Vehicle 1: 0→4→5→0;

Vehicle 2: 0→3→7→1→0;

Vehicle 3: 0→2→6→8→0.

4.2. Crossover operation

This paper presents a crossover manner, which is improved on the CX manner [12], in order to retain the good gene fragments of parents. Suppose two parent individuals are *f*1 and *f*2, through crossover operation to be the offspring individuals *new*1 and *new*2. Specific operation such as：

Step 1: Randomly select one cross region (a, b) in the second string;

Step 2: Put the number in the cross region of *f*2 to the front of *f*1, delete the number of the cross region of *f*2 in *f*1, get a sub-string *new*1. Put the number in the cross region of *f*1 to the front of *f*2, delete the number of the cross region of *f*1 in *f*2, get a sub-string *new*2;

Step 3: Judge whether the sub-string is a feasible solution, if not, redo **Step 2**, otherwise go to **Step 4**;

Step 4: Calculate the fitness by Eq. (1), retain the offspring of the lower fitness recorded as *new*.

For example, given two parent strings:

*f*1=1 2 3 4 5 6 7 8 9, *f*2=9 8 7 6 5 4 3 2 1,

and the cross region (3, 6), after crossing, two new strings are:

*new*1=7 6 5 4 1 2 3 8 9, *new*2=3 4 5 6 9 8 7 2 1.

Calculate the fitness for *new*1 and *new*2 by Eq. (1). Lastly, crossover operation of two parent strings is completed. Obviously, owing to crossover operation, the number of particles become twice as many as that of the original one, namely 2N new particles are obtained.

4.3. Selection operation

We know that in PSO algorithm, each particle moves towards the global optimal location and the optimal location of individual history as a criteria to find a better location for survival. However, this movement in PSO makes some inferior particles continue to reproduce to become inferior communities unable to be eliminated which affects the algorithm convergence rate. At the same time the resource of particle swam can not be fully utilized. So in selection operation, we need to judge which particles will be eliminated and which will be chosed to next generation. This requires that all particles should be divided into two parts: the superior ones and the inferior ones. Level set theory is introduced here.

For the tth-generation $P(t) = (P_1, P_2, ..., P_n)$, n denots the number of particles, the fitness function of particles is set to $f_i(x)$, order $\overline{f_t} = \sum_{i=1}^{n} \frac{f(X_i)}{n}$, where t denotes tth-generation.

We have $H_{\bar{f}_t} = \{x_i \in P(t) \mid f(x_i) \le \overline{f}_t, 1, 2, \cdots, n\}$

$H_{\bar{f}_t}$ is called the level set about f relative to $P(t)$. After that the population of each generation can be divided two parts: $H_{\bar{f}_t}$ and $P(t) - H_{\bar{f}_t}$. [8] In the process of Crossover operation, $2N$ new particles were obtained, while only N of them can be allowed to enter next generation. Thus, we sort the $2N$ particles on their fitness measured by Eq. (1) in descending order, and the top N particles would be selected to enter the next generation.

4.4. Algorithm implementation

In this paper, we use the hybrid PSO to solve VRPTW, whose pseudocode of algorithm is described in Fig 1:

Set iteration times for *MaxN*, randomly generated initial particles *N*;
Calculate the fitness of initial particle to be l_0. According to the initial fitness of each particle, initialize $plbest_i$, $pxbest_i$, *glbest* and *gxbest*;
WHILE (iteration times< *MaxN*) DO
    FOR i=1: *MaxN*
        The ith particle X_i crosses with *gxbest* and selection to be $X_i^{'}$; $X_i^{'}$ crosses with $pxbest_i$ and selection to be $X_i^{''}$, ordering $X_i=X_i^{''}$;
        Calculate fitness l_i according to current location;
        IF ($l_i < plbest_i$)
            $pxbest_i=X_i$, $plbest_i=l_i$;
        END IF
    END FOR
    Update $plbest_i$, $pxbest_i$, *glbest*, *gxbest*;
END WHILE
Finally, export *glbest* and *gxbest*.

Figure. 1. Pseudocode of hybrid PSO algorithm to solve VRPTW

5. Calculation example

In this paper, we use the example in literature [8]. There are eight cargo transportation tasks (No. 1, 2,…,8). The cargo q_i of each task, loading or unloading time t_i, executing time range [ET_i, LT_i] are showed in Tab I, distance d_{ij} from yard 0 to each task point in Tab II. These tasks are completed by eight tons of vehicles, in which the speed is 50, unit transportation cost is 1, over time window unit cost for punishment is: P_E=P_L=50.

TABLE I. CHARACTERS AND DEMAND OF TASK

Task point	1	2	3	4	5	6	7	8
q_i	2	1.5	4.5	3	1.5	4	2.5	3
t_i	1	2	1	3	2	2.5	3	0.8
[ET_i, LT_i]	[1,4]	[4,6]	[1,2]	[4,7]	[3,5.5]	[2,5]	[5,8]	[1.5,4]

TABLE II. DISTANCE FROM POINT TO POINT

d_{ij}	0	1	2	3	4	5	6	7	8
0	0	40	60	75	90	200	100	160	80
1	40	0	65	40	100	50	75	110	100
2	60	65	0	75	100	100	75	75	75
3	75	40	75	0	100	50	90	90	150
4	90	100	100	100	0	100	75	75	100
5	200	50	100	50	100	0	70	90	75
6	100	75	75	90	75	70	0	70	100
7	160	110	75	90	75	90	70	0	100
8	80	100	75	150	100	75	100	100	0

The best known result is: vehicle 1: 0→3→1→2→0; vehicle 2: 0→8→5→7→0; vehicle 3: 0→6→4→0. The total travel cost is Z=910.

Set parameters of the Hybrid-PSO as follows: the number of particles N=40, the iteration time $MaxN$=200.Solving this problem uses GA in literature [8] (N=40, $MaxN$=200, Pc=0.6, Pe=0.2 roulette manner choosing offspring); uses PSO in literature [13] (N=40, divided into two subgroups, subgroup size=22, overlapping particle swarm between subgroups is 2, w=0.729, c_1=c_2=1.49445, $MaxN$=200); uses parallel PSO in literature [11] (divided into two subgroups, subgroup size is $N1$=$N2$=20, w=0.729, c_1=c_2=1.49445, $MaxN$=200). These above three methods and Hybrid-PSO algorithm use Matlab7.0 to program respectively, each with the operator of 50 times. The comparison results are shown in Tab III:

TABLE III COMPARED RESULTS IN 4 ALGORITHMS FOR SOLVING 50 TIMES

	Average travel cost	Searching successful rate (%)	Average searching time (s)
GA[8]	993.6	24	11
PSO[13]	940.5	46	6
Parallel PSO[11]	923.8	72	4
Hybrid-PSO	914	97	4

Experimental results show that the searching successful rate 97% of the proposed Hybrid-PSO in solving VRPTW problem is far greater than the basic

PSO 46%, the basic GA 24% and 72% of parallel PSO. At the same time in the aspect of the average result and evaluating successful searching time, the proposed method compared with other three methods is optimal.

6. Conclusion

This paper presents a HPSO algorithm, which combines the PSO algorithm with genetic algorithm and level set. Through an example about window time VRP validated, the algorithm which has the characters of less number of parameters, simple implementation, and relatively shorter time of average search, is proved as an effective algorithm for solving discrete optimization problems.

7. Acknowledgment

The authors thank the anonymous reviewers for their valuable comments and suggestions that helped us for improving our work.

8. References

[1] Dantzig G, Ramser J. "The truck dispatching problem". *Management Science*, 1959(6), pp. 80-91.

[2] Olli Braysy. "Genetic algorithms for the vehicle routing problem with time windows". *Special issue on Bioinformatics and Genetic Algorithms*, 2001(1), pp. 33–38.

[3] Hoong C L, Melvyn S, Kwong M T. "Vehicle routing problem with time windows and a limited number of vehicles". *European Journal of Operational Research*, 2003, 148: pp.559-569.

[4] R C Eberhart, J Kennedy. "A New Optimizer Using Particles Swarm Theory". *In: Proc Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995.

[5] DING Hai-Li, WANG Fang, GAO Cheng-xiu. "Crossover particle swarm optimization for traveling salesman problem". *Journal of Mathematics*, 2008(8): pp. 85-89.

[6] Niu, B., Zhu, Y.L. and He, X.X., et al. An Improved Particle Swarm Optimization Based on Bacterial Chemotaxis, *In Proc. IEEE Congress on Intelligent Control,* Dalian, CHINA, 2006, pp.3193-3197.

[7] Jiang Wei-gang, Zhang Yuan-biao, Xie Jian-wen. "A Particle Swarm Optimization Algorithm Based on Diffusion-Repulsion and Application to Portfolio Selection," *2008 International Symposium on Information Science and Engineering*, 2008(2): pp. 498-501

[8] LI Jun,GUO Yao-huang. "Dispatch optimizes theory and methods for logistic distribution vehicle". Beijing: Chinese Commodity Publish House, 2001.

[9] Maurice Clerc, James Kennedy. "The particle swarm-explosion, stability, and convergence in a multidimensional complex space". *IEEE Transaction on Evolutionary Computation*, 2002, 6(1), pp. 58-73.

[10] ZHANG Li-yan, PANG Xiao-hong, XIA Wei-jun, et al. "A hybrid particle swarm optimization algorithm for vehicle routing problem with time windows". *Journal of Shanghai Jiaotong University*, 2006(11), pp. 1890-1895.

[11] WU Yong, YE Chun-ming, MA Hui-min, et al. "Parallel particle swarm optimization algorithm for vehicle routing problems with time windows". *Computer Engineering and Applications*, 2007, 43(14), pp. 223-226.

[12] ZHANG Li-ping, CHAI Yue-ting, CAO Rui. "Improved genetic algorithm for vehicle routing problem with time windows". *Computer Integrated Manufacturing Systems*, 2002, 8(6), pp. 451-454.

[13] LI Ning, ZOU Tong, SUN De-bao. "Particle swarm optimization for vehicle routing problem with time windows". *Systems Engineering-Theory & Practices*, 2004(4), pp. 130-135.