International Journal of u- and e- Service, Science and Technology
Vol.9, No. 8 (2016), pp.97-110
http://dx.doi.org/10.14257/ijunesst.2016.9.8.09

Memetic Two-echelon Vehicle Routing Optimization Based on Q Learning Theory and Differential Evolution Algorithm

Liu Dongdong[1,a], Liu Kai1, Wang Feng[1], Han Bo[1], Zhao Zhengping[1], Tan Fuxiao[1] and Niu Lei[1]

[1]*School of Computer and Information Engineering, Fuyang Normal University, Fuyang, Anhui Province, China, 236041*
a.ourliudd@163.com

Abstract

In allusion to such problems as low accuracy and long convergence time in traditional two-echelon vehicle routing optimization algorithm, a Memetic algorithm (QDEMA) based on Q learning theory and differential evolution is proposed in this article to solve above problems. Firstly, it is necessary to research the two-echelon vehicle routing optimization problem and adopt the optimal segmentation method to obtain the relatively reasonable distribution plan for the first-echelon SDVRP problem in order to accordingly determine the distribution quantity of the transfer stations; secondly, the second-echelon MDVRP distribution scheme is solved to obtain the total distance and the total number of the distribution vehicles for the two-echelon optimization problem; thirdly, in allusion to the solution of the second-echelon MDVRP distribution scheme, Q learning theory and the differential evaluation algorithm are adopted to design new Memetic algorithm in order to globally optimize MDVRP distribution scheme; finally, the simulation experiment is carried out to verify the algorithm effectiveness.

Keywords: *Q Learning; Differential Evolution; Memetic; Two-echelon Vehicle Routing Optimization*

1. Introduction

At present, the research on vehicle routing problem (VRP) is mainly focused on the single-echelon mode, namely the single-echelon distribution mode for delivering the cargos from the central warehouse to users [1]. However, the number of the roads which trucks are restricted to travel on is increased along with the increase of the urban traffic pressure. As a result, the single-echelon distribution mode gradually fails to satisfy the distribution requirement, and the common solution is to establish the transfer stations to take charge of the two-echelon distribution for customers.

The two-echelon vehicle routing optimization problem (2E-VRP) is different from VRP in the aspects of solving mode and difficulty. Specifically, the first-echelon distribution and the second-echelon distribution are coupled in 2E-VRP, and the traditional optimization model and algorithm have poor accuracy and long calculation time. Scholars' research on 2E-VRP is mainly focused on two directions accurate optimization and heuristic optimization. For example, the branch cut algorithm is adopted in the commodity flow model of the two-echelon vehicle optimization problem in literature [2], and this algorithm has high precision but long operation time; the above branch cut algorithm is improved in literature [3] in order to further improve the algorithm performance; the inequation method is adopted in literature [4] to strengthen the continuity and the slackness, *etc.*, of the two-echelon vehicle routing optimization problem in the branch cut algorithm.

ISSN: 2005-4246 IJUNESST

The accurate algorithm has high accuracy but long operation time, thus inapplicable to large distribution network. Therefore, the heuristic algorithm is generated at the right moment. For example, the classification planning parallel heuristic algorithm is proposed in literature [5] to divide the two-echelon vehicle routing optimization problem into two vehicle distribution sub-problems, and the planning algorithm is adopted to solve the two sub-problems; although the search speed is accelerated, yet the solution accuracy is still dissatisfactory due to the limitation to the coupling performance and the optimization algorithm performance.

Based on the division mode in literature [5], the accurate optimal cut algorithm is adopted in this article for the first-echelon distribution to lay a foundation for the second-echelon optimization, and the improved QDEMA algorithm is adopted for the second-echelon optimization for global optimization, thus to solve the coupling problem of the two-echelon algorithm and meanwhile reduce the algorithm operation time.

2. QDEMA Two-Echelon Vehicle Routing Problem

2.1. Problem Description

The two-echelon vehicle routing optimization problem (2E-VRP) is as shown in Figure 1, wherein $S_1 \sim S_3$ are the first-echelon transfer stations, $c_1 \sim c_9$ are the second-echelon user points, and the two-echelon vehicle routes are distinguished by different lines (full line for the first-echelon and imaginary line for the second-echelon).

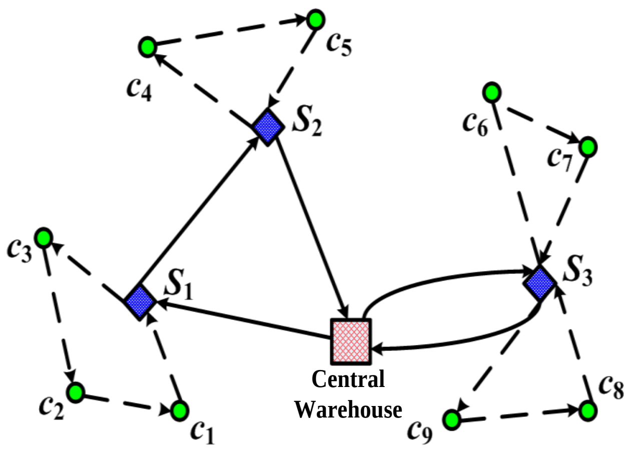


Figure 1. 2E-VRP Problem

2E-VRP mathematical model is equivalent to the minimum distribution time problem, namely: the total time till the last vehicle finishes the distribution shall be minimized[6]:

$$f = \min(C_{max}) \tag{1}$$

Constraint (2) is adopted to ensure that each order is only processed once;

$$\sum_{k=1}^{n} Y_{ik} = 1, i = 1, 2, \cdots, n \tag{2}$$

Constraint (3) is adopted to ensure that if $\sum_{i=1}^{n} Y_{ik} = 0, k = 1, 2, \cdots, n-1$ is true, then $\sum_{i=1}^{n} Y_{i,k+1} \neq 1$ is true, namely: if there is no order number k , then there will be no order number $k+1$;

$$\sum_{i=1}^{n} Y_{i,k+1} \leq H \sum_{i=1}^{n} Y_{ik} \tag{3}$$

Constraint (4) is adopted to ensure that if $\sum_{i=1}^{n} Y_{ik} = 0$ is true, then $y_k = 0$ is true;

$$y_k \leq \sum_{i=1}^{n} Y_{ik}, \quad k = 1, \cdots, n \tag{4}$$

Constraint (5) is adopted to ensure that if $\sum_{i=1}^{n} Y_{ik} \geq 1$ is true, then $y_k = 1$ is true;

$$\sum_{i=1}^{n} Y_{ik} \leq H \times y_k, \quad k = 1, \cdots, n \tag{5}$$

Constraints (4) and (5): if order B_k is valid, then $y_k = 1$ is true; if order B_k is invalid, then $y_k = 0$ is true.

Constraints (6)~(8) are adopted to give the distribution vehicle batch limitation;

$$\sum_{g=1}^{G} V_{kg} \leq 1, \ k = 1, \cdots, n \tag{6}$$

$$\sum_{g=1}^{G} V_{kg} \leq H \times y_k, \ k = 1, \cdots, n \tag{7}$$

$$\sum_{k=1}^{n} \sum_{g=1}^{G} V_{kg} = \sum_{k=1}^{n} y_k \tag{8}$$

Constraint (6) is adopted to limit each order to be delivered only by one vehicle; Constraint (7) indicates that currentt customer does not have any order and there is no need to arrange distribution; Constraint (9) is adopted to give vehicle capacity limitation;

$$\sum_{i=1}^{n} e_i Y_{ik} \leq z_g + H\left(1 - V_{kg}\right), \ k = 1, \cdots, n, g = 1, \cdots, G \tag{9}$$

Constraints (10) and (11) are adopted to give the preparation time for batch B_k;

$$r_k \geq \sum_{k'=1}^{k} \sum_{i=1}^{n} p_i Y_{ik'} - H\left(1 - y_k\right), k = 1, \cdots, n \tag{10}$$

$$r_k \leq \sum_{k'=1}^{k} \sum_{i=1}^{n} p_i Y_{ik'} - H\left(1 - y_k\right), k = 1, \cdots, n \tag{11}$$

Constraints (12)~914) are adopted to give the starting time u_k limitation;

$$u_k \leq H \times y_k, k = 1, \cdots, n \tag{12}$$

$$\begin{cases} u_1 = r_1 \\ u_k \geq r_k, \end{cases} k = 2, \cdots, n \tag{13}$$

$$\begin{cases} u_{k'} + t_{01}^{g} + t_{10}^{g} - H\left(2 - V_{kg} - V_{k'g}\right) \le u_{k} \le \\ \quad u_{k'} + t_{01}^{g} + t_{10}^{g} - H\left(2 - V_{kg} - V_{k'g}\right) \\ k = 2,\cdots,n;\ k' = 1,\cdots,k-1;\ g = 1,\cdots,G \end{cases} \tag{14}$$

Constraint (15) is adopted to give the limitation condition of the order distribution time T_k .

$$\begin{cases} T_{k} \le H \times y_{k}, k = 1,\cdots,n \\ u_{k} + t_{01}^{g} + t_{10}^{g} - H\left(1 - V_{kg}\right) \le u_{k} \le \\ \quad u_{k} + t_{01}^{g} + t_{10}^{g} - H\left(1 - V_{kg}\right) \\ k = 2,\cdots,n;\ g = 1,\cdots,G \end{cases} \tag{15}$$

2.2. Encoding Mode and Initial Cluster

2E-VRP can be regarded as the coupling problem of first-echelon SDVRP and second-echelon MDVRP and can be decoupled according to the order from the second-echelon to the first-echelon. Firstly, the optimal cut algorithm is adopted to obtain the relatively reasonable distribution plane of the first-echelon SDVRP in order to determine the number of the transfer stations and meanwhile take it as the initial solution of QDEMA algorithm. Secondly, the two-echelon MDVRP distribution scheme is solved to obtain the total distance and the total number of distribution vehicles. Therein, QDEMA algorithm aims at only optimizing the second-echelon customer distribution scheme, thus to be favorable for improving the algorithm efficiency. The optimal cut algorithm process is as shown in Figure 2, [7].

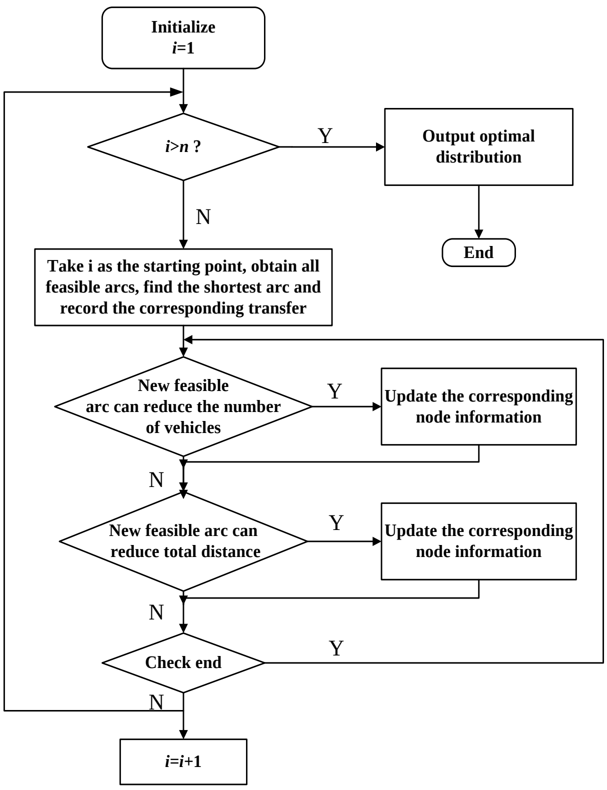


Figure 2. Optimal Cut Algorithm

3. Q Learning Theory and DE Algorithm

3.1. Differential Evolution Algorithm

Due to simple structure, rapid convergence and high accuracy, DE (Differential Evolution) algorithm is widely applied. If DE algorithm has NP cluster vectors, then the Gth cluster can be expressed as follows[8]:

$$P_G = \{\mathbf{X}_1(G), \mathbf{X}_2(G), \cdots, \mathbf{X}_{NP}(G)\} \tag{16}$$

In the above formula, $\mathbf{X}_i(G), i \in 1, \cdots, NP$ denotes the cluster individual. DE algorithm can be implemented through the following operations[9]:

Step 1: (Initialization) For the cluster ($G = 0$), the individual $\mathbf{X}_i(0)$ thereof can be randomly realized in the value range $[\mathbf{X}_{\min}, \mathbf{X}_{\max}]$:

$$\begin{cases} \mathbf{X}_{\min} = \{x_{\min-1}, \cdots, x_{\min-D}\} \\ \mathbf{X}_{\max} = \{x_{\max-1}, \cdots, x_{\max-D}\} \end{cases} \tag{17}$$

In the above formula, D is the DE cluster dimensionality. The j th element of the individual i in the cluster ($G = 0$) can be initialized as follows:

$$x_{ij}(0) = x_{\min-j} + rand_{ij}(0,1) \times \left(x_{\max-j} - x_{\min-j}\right) \tag{18}$$

In the above formula, $rand_{ij}(0,1)$ is a uniform distribution function in the interval $[0,1]$ and can be used to randomly initialize crossover probability factor Cr .

Step 2: (Mutation) Standard DE mutation is as follows: randomly select 2 clusters ($\mathbf{X}_{rand-1}(G)$, $\mathbf{X}_{rand-2}(G)$) to generate a new individual $\mathbf{V}_i(G)$ through the vector superposition with the target individual $\mathbf{X}_i(G)$. Namely:

$$\begin{aligned} \mathbf{V}_i(G) = \mathbf{X}_i(G) + F_1\left(\mathbf{X}_{best}(G) - \mathbf{X}_i(G)\right) \\ + F_2\left(\mathbf{X}_{rand-1}(G) - \mathbf{X}_{rand-2}(G)\right) \end{aligned} \tag{19}$$

In the above formula, F ($F \in [0,2]$) is a proportional factor. The relatively simple mutation method is adopted (as shown in Formula (4)).

Step 3: (Crossover) There are two types of crossover modes, namely binomial crossover and exponential crossover:

Binomial crossover: implement the crossover operation to donator vector $\mathbf{V}_i(G)$ and target vector $\mathbf{X}_i(G)$ to generate a new individual $\mathbf{U}_i(G)$ as follows:

$$u_{ij}(G) = \begin{cases} v_{ij}(G), if\ rand_{ij} \le Cr\ or\ j = j_{rand} \\ x_{ij}(G), otherwise \end{cases} \tag{20}$$

Exponential crossover: randomly select an integer n in the interval $[1, D]$ as the starting point for the target vector $\mathbf{X}_i(G)$ to start the element exchange with the donator vector $\mathbf{V}_i(G)$. Similarly, randomly select another integer L in the interval $[1, D]$ as the number of the elements which are contributed by the donator vector to the new individual vector. The exponential crossover is as follows:

$$u_{ij}(G)=\begin{cases} v_{ij}(G),\ for\ j=\langle n\rangle_D,\cdots,\langle n+L-1\rangle_D \\ x_{ij}(G), otherwise \end{cases} \tag{21}$$

In the above formula, $\langle\cdot\rangle_D$ denotes the modular function with the modulus as D .

Step 4: (Selection) For the minimization problem with given target function $f(\mathbf{x})$, the elitist selection mode of DE algorithm can be expressed as follows:

$$\mathbf{X}_i(G+1)=\begin{cases} \mathbf{U}_i(G), if\ f(\mathbf{U}_i(G))\le f(\mathbf{X}_i(G)) \\ \mathbf{X}_i(G), if\ f(\mathbf{U}_i(G))> f(\mathbf{X}_i(G)) \end{cases} \tag{22}$$

3.2. Q Learning Theory

Q learning is a reinforcement learning method [10], wherein the corresponding award (punishment) is given to the operation that can be used to change the environment state so as to make the operation implemented towards the clear target direction. In practical application, it is very difficult to predict the award provided to future state s' in state s . Only the optimal action award of the future state s' is considered in Q learning.

The parameters are set as follows: $S=\{s_1,\cdots,s_n\}$ is the state set of the intelligent agents in a given environment; $A=\{a_1,\cdots,a_n\}$ is the set of the selectable actions in the state $s_i\in S$ for the intelligent agents; $r(s_i,a_j)$ is the immediate award given to the intelligent agent for selecting action a_j in state s_i; $\delta(s_i,a_j)$ is the transition function of s_k which is the next state after the intelligent agent selects action a_j in state s_i; γ is the discount factor for the punishment of the future award delay, $\gamma\in[0,1]$; $Q(s_i,a_j)$ is the total awards given to the intelligent agent for selecting action a_j in state s_i. Then, $Q(s,a)$ can be expressed as follows [11]:

$$\begin{aligned} Q(s,a)&=r(s,a)+\gamma \overset{*}{V}\,Q(\delta(s,a)) \\ &=r(s,a)+\gamma\max_{a'}Q(\delta(s,a),a') \end{aligned} \tag{23}$$

In the above formula, $\overset{*}{V}$ is the total awards obtained by the intelligent agent in state s .

The differential improvement form of Q learning algorithm is given as follows:

$$\begin{aligned} Q(s,a)\leftarrow &(1-\alpha)Q(s,a)+ \\ &\alpha\times\left(r(s,a)+\gamma\max_{a'}Q(\delta(s,a),a')\right) \end{aligned} \tag{24}$$

In the above formula, when action a points to $\delta(s,a)$, Q-value of $Q(s,a)$ shall be progressively increased in order to ensure that the award $r(s,a)$ for the next action is more than $Q(s,a)$, thus to promote the evolution towards the optimal direction. When $\alpha=0$ is true, the intelligent agent stops learning; when $\alpha=1$ is true, the intelligent agent only considers the newest information. The discount factor γ is used to determine the importance of the future information, wherein $\gamma=0$

indicates that the algorithm focuses on current award and $\gamma = 1$ indicates that the algorithm focuses on the long-term great awards.

4. QDEMA Algorithm

The differential evolution algorithm is adopted for QDEMA algorithm to realize the global optimization, and DQL (Differential Q Learning) algorithm is adopted to realize local deep exploration. The pseudo codes of QDEMA algorithm are as shown in Table 1 and the algorithm steps are as follows:

Step 1: (Initialization) In the initial search range, initialize the cluster with the size as NP and the dimensionality as D, wherein the selection method is similar to Formula (3), initialize Q-table as a relatively small value, and if Q-table is maximally as 100, then assign 1 to the corresponding Q-table value;

Step 2: (Parameter Adaption) Mainly adopt the award and punishment measures of Q-table to select the proportional factor F suitable for the algorithm, and calculate the probability of selecting $F = F_j$ according to the following formula:

$$P\left(F_j\right) = Q\left(s_i, 10F_j\right) \Big/ \sum_{l=1}^{10} Q\left(s_i, 10F_l\right) \tag{25}$$

In order to ensure the self-adaption of Q-value in each row, it is necessary to randomly generate a numerical value r in the interval $[0,1]$ and then select F_j, wherein the following conditions shall be met:

$$\begin{aligned} &\sum_{m=1}^{j-1} P\left(F = F_m\right) < r \le \sum_{m=1}^{j} P\left(F = F_m\right) \\ &\Rightarrow \frac{\sum_{m=1}^{j-1} Q\left(s_i, 10F_m\right)}{\sum_{l=1}^{10} Q\left(s_i, 10F_l\right)} < 1 \le \frac{\sum_{m=1}^{j} Q\left(s_i, 10F_m\right)}{\sum_{l=1}^{10} Q\left(s_i, 10F_l\right)} \end{aligned} \tag{26}$$

Step 3: (DE Operation) Adopt DE algorithm for individual ordering and state allocation, wherein f_i is the newest target value of individual i, normalize f_i through $f_i \Big/ \sum_{j=1}^{NP} f_i$, and rank the normalized f_i by a descending order to obtain the ranked list with the rank as r and the state as s_r; then, repeat the above operations for $r = 1 : NP$;

Step 4: (Q-table Updating) If the state of the individual is changed from original state s_i to state s_k after executing the operation F_j and the target adaption value is increased, then it is necessary to adopt the positive award formula to update $Q\left(s_i, 10F_j\right)$ as follows:

$$\begin{aligned} Q\left(s_i, 10F_j\right) &= (1-\alpha) Q\left(s_i, 10F_j\right) + \\ &\alpha\left(reward\left(s_i, 10F_j\right) + \gamma \max_{F'}\left(s_k, 10F'\right)\right) \end{aligned} \tag{27}$$

Or else, it is necessary to adopt the negative award $-K$ to update $Q\left(s_i, 10F_j\right)$.

Step 5: (Convergence Judgment) Repeat steps 2~4 till the following condition is met: reach the iteration termination number or satisfy the convergence accuracy requirement.

Table 1. Pseudo Codes of QDEMA Algorithm

1. //**Algorithm initialization**

2. Set iteration number $t = 0$, initialize NP cluster $P_0 = \{\mathbf{X}_1(0), \cdots, \mathbf{X}_{NP}(0)\}$. Set: $\alpha = 0.25$ and $\gamma = 0.8$, calculate the adaption value $f(\mathbf{X}_i(0))$ of $\mathbf{X}_i(0)$ and rank it. $R(0) = [r_1(0), \cdots, r_{NP}(0)]$, wherein $r_i(0)$ is the target vector with the rank as *i* at the t[th] generation. Initialize $Q(r_i(0), j) = 1$.

3. ***while*** dissatisfy the termination condition ***do***

Set $[reward(r_i(t), j)] = 0$, $r_i = [1, \cdots, NP]$, $j = 1:10$;

for $i = 1:NP$ ***do***

4. //**Roulette selection**

Randomly select the proportional factor F_{r_i} in the interval $[0.1,1]$, with the selection probability as: $P(F = F_j) = Q(r_i(t), j) \Big/ \sum_{l=1}^{10} Q(r_i(t), l)$;

5. //**Mutation operation**

Execute the mutation operation to the i^{th} target vector $\mathbf{X}_i(t)$:

$\mathbf{V}_i(t) = \mathbf{X}_i(t) + F(\mathbf{X}_{best}(t) - \mathbf{X}_i(t)) + F(\mathbf{X}_{rand1}(t) - \mathbf{X}_{rand2}(t))$

6. //**Crossover operation**

Adopt Formula (20) or Formula (21) to execute the crossover operation to the target vector $\mathbf{X}_i(t)$ and the mutated vector $\mathbf{V}_i(t)$ to generate $\mathbf{U}_i(t)$;

7. //**Selection operation**

if $f(\mathbf{U}_i(t)) < f(\mathbf{X}_i(t))$

$reward(r_i(t), 10F_{r_i}) = f(\mathbf{X}_i(t) - \mathbf{U}_i(t))$; $\mathbf{X}_i(t+1) = \mathbf{U}_i(t)$;

if $f(\mathbf{U}_i(t)) < f(\mathbf{X}_{best}(t))$

$\mathbf{X}_{best}(t) = \mathbf{U}_i(t)$ **; update** $\mathbf{X}_{best}(t)$;

endif

else

$reward(r_i(t), 10F_{r_i}) = -K$; $\mathbf{X}_i(t+1) = \mathbf{X}_i(t)$;

endif

Update $\mathbf{X}_{best}(t)$;

endfor

8. Rank the cluster by a descending order: $R(t+1) = [r_1(t+1), \cdots, r_{NP}(t+1)]$;

9. //**Update Q-table**

for $i = 1:NP$

for $F_j = 0.1:1.0$

if $reward(r_i(t), 10F_j) \neq 0$

$$Q(s_i, 10F_j) = (1-\alpha)Q(s_i, 10F_j) + \alpha\left(reward(s_i, 10F_j) + \gamma \max_{F'}(s_k, 10F')\right);$$

else

$Q(s_i, 10F_j) = Q(s_i, 10F_j)$;

endif

endfor

endfor

$t = t + 1$;

10. ***endwhile***

5. Simulation Experiment and Analysis

5.1. QDEMA Algorithm Experiment

Test functions: $f1$ is Dejong and $f2$ is Griewank. The global optimal value of functions $f1$ and $f2$ is 0. The above two functions are as follows:

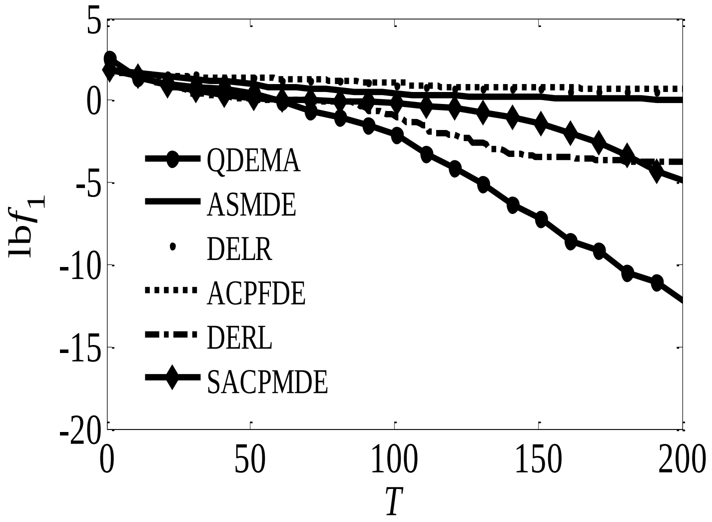
$$f1 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(x_i / \sqrt{i}\right) + 1,\ [-60,60]$$

$$f2 = \frac{\sin^2\left(\sqrt{x_1^2 + x_2^2}\right) - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2} - 0.5,\ [-100,100]$$

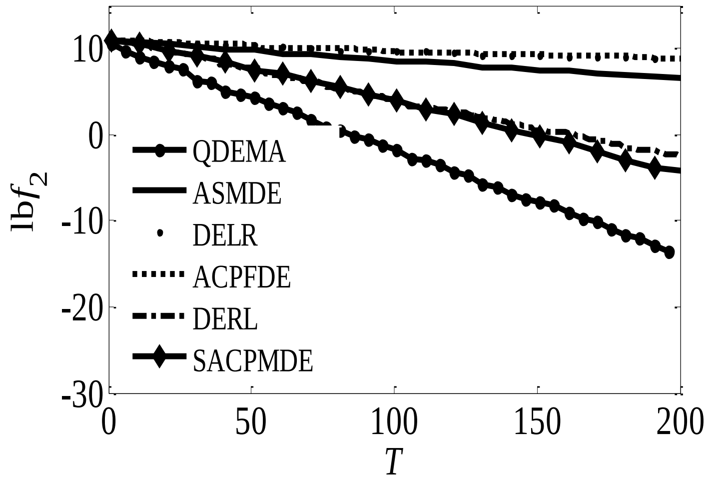
The following five algorithms ACPFDE, ASMDE, DERL, SACPMDE and DELR are selected as the comparison algorithms. According to the algorithm parameter setting method in relevant literatures, the algorithm parameters are set as follows: $D = 30$, cluster size $NP = 200$, iteration termination number = 8,000.

The value range of the crossover probability factor of ACPFDE algorithm is set as $CR \in [0.3, 0.9]$, the parameter settings of SACPMDE algorithm and ASMDE algorithm are as the same as relevant setting in literature [12], the parameter settings of DERL algorithm and DELR algorithm are as the same as relevant setting in literature [13], the crossover probability factor of the above four algorithms is set as $CR = 0.9$, and the simulation accuracy is set as *VTR*=10-6. The simulation comparison results of the comparison algorithms are as shown in Figures 3, (a) and 3(b)

Figures 3, (a) and 3, (b) respectively show the simulation comparison results of the above algorithms on Dejong and Griewank test functions. According to Figure 3, (a), the above algorithms present two different convergence tendency: firstly, QDEMA algorithm and SACPMDE algorithm does not present premature convergence, but the convergence rate of SACPMDE algorithm is obviously less than that of QDEMA algorithm; secondly, the other algorithms more or less present premature convergence. According to Figure 3, (b), the above algorithms does not present any premature convergence, but the convergence rate of QDEMA algorithm is obviously more than those of the comparison algorithms. The simulation on the test functions can more or less verify the effectiveness of the proposed algorithm.



(a) Dejong Simulation Comparison



(b) Griewank Simulation Comparison

Figure 3. Test Function Simulation Comparison

4.2. Calculation Example Experiment

Experiment condition setting: the calculation example set (set2) in literature [1-2] is adopted as the calculation example; hardware configuration: 2.60GHz processor, 4GB memory, win7 system; simulation software: matlab2012b; experiment parameter setting: differential evolution cluster size $NP = 50$, crossover probability factor $P_c = 0.8$, iteration termination times $T_{\max} = 1000$, maximum run time of algorithm $t_{\max} = 60$.

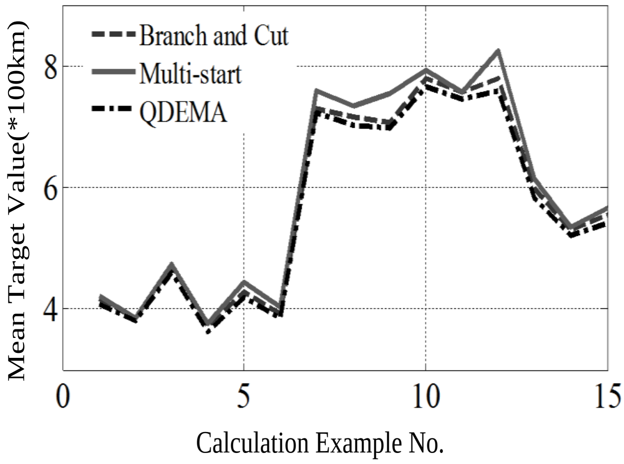
Simulation comparison index: the target value of the best distribution scheme (*best*) and the mean run time (*time*) are obtained after the algorithm runs for 10 times. Branch and Cut [14] algorithm and Multi-start [15] algorithm are adopted as the comparison algorithms. The simulation results are as shown in Table 2 and Figure 4.

Table 2. Simulation Comparison Result

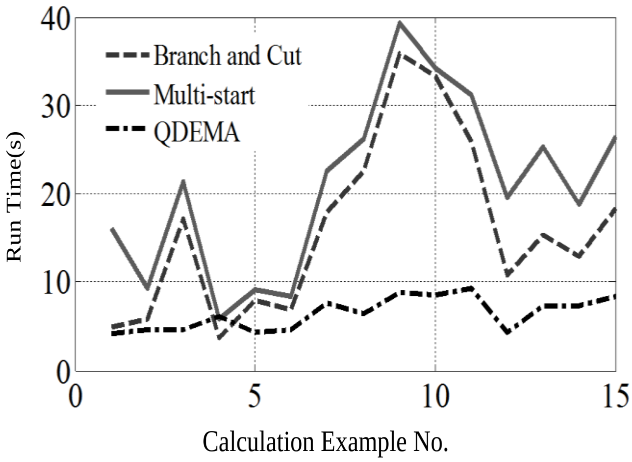
Calculation example	Branch and Cut		Multi-start		QDEMA	
	Best	Time	Best	Time	Best	Time
E-n22-k4-s6-17	417	4.94	421	16.12	407	4.21
E-n22-k4-s8-14	384	5.76	384	9.23	379	4.58
E-n22-k4-s9-20	481	17.19	472	21.36	458	4.65
E-n22-k4-s10-14	371	3.64	375	5.87	362	6.07
E-n22-k4-s11-12	427	7.87	444	9.15	418	4.26
E-n22-k4-s12-16	392	6.82	403	8.34	384	4.67
E-n33-k4-s1-9	730	17.92	757	22.64	721	7.64
E-n33-k4-s2-13	714	22.52	733	26.14	702	6.46
E-n33-k4-s3-17	707	35.8	754	39.28	698	8.84
E-n33-k4-s4-5	778	33.25	792	34.18	765	8.51
E-n33-k4-s7-25	756	25.87	756	31.24	745	9.30
E-n33-k4-s14-22	779	10.75	824	19.57	758	4.37
E-n51-k5-s2-17	597	15.27	614	25.31	581	7.36
E-n51-k5-s4-46	530	12.87	533	18.74	521	7.26
E-n51-k5-s6-12	554	18.34	564	26.47	541	8.40

The simulation data of the three comparison algorithms on different calculation examples are shown in Table 2, wherein the unit of *Best* is kM and unit of *Time* is s. The comparisons of the mean target values and the run time of the three algorithms are as shown in Figures 4, (a) and 4, (b). According to Figure 4, (a), although the three algorithms have approximate optimal values, QDEMA algorithm is always superior to Branch and Cut algorithm and Multi-start algorithm, and Multi-start algorithm is always superior to Branch and Cut algorithm. In the aspect of run time, Multi-start algorithm has longest run time, Branch and Cut algorithm has intermediate run time, and QDEMA algorithm has shortest run time. Compared with Branch and Cut algorithm, Multi-start algorithm has improved algorithm convergence accuracy but long computing time. QDEMA algorithm is superior to other two comparison algorithms in the aspects of convergence accuracy and run time, and has stable mean run time.

In extremely specific condition, for example, in simple calculation example 4, Branch and Cut algorithm and Multi-start algorithm have short rum time which is even approximate to or shorter than the run time of QDEMA algorithm, because the above calculation example is too simple but QDEMA algorithm still adopts the cluster optimization method which takes long time for computation.



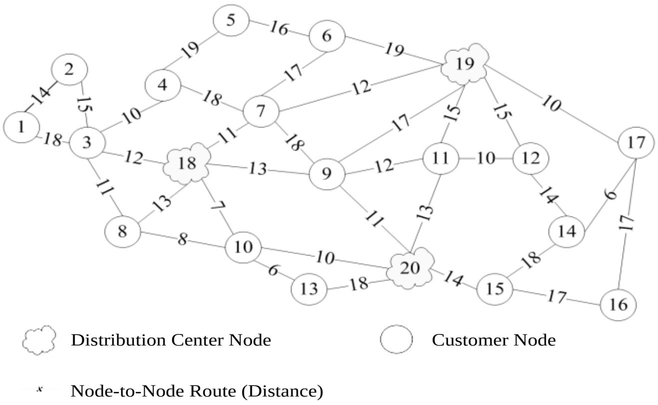
(a) Optimal Value Comparison Result



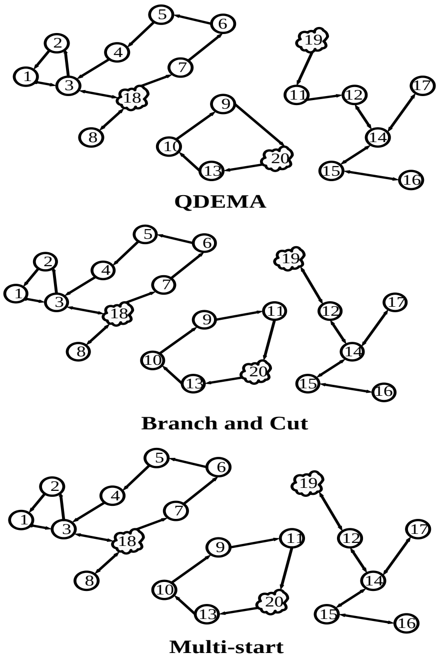
(b) Mean Run Time

Figure 4. Algorithm Comparison Result

E-n22-k4-s9-20 calculation example is taken as an example to give the distribution routes calculated by QDEMA algorithm, Branch and Cut algorithm and Multi-start algorithm, as shown in Figures 5, (a)~(b). Therein, Figure (a) is the original node distance diagram, and the distribution distance (unit: kM) between the stations is also shown in this figure. Figure, (b) shows the distribution route diagrams of the above three algorithms. According to the simulation comparison results shown in Table 2, the total distances of the best distribution routes of QDEMA algorithm, Branch and Cut algorithm and Multi-start algorithm are respectively 458kM, 470kM and 472kM. Obviously, QDEMA algorithm is superior to the comparison algorithms. Meanwhile, QDEMA algorithm and Branch and Cut algorithm are about the same in the aspect of run time, but are superior to Multi-start algorithm. The distribution schemes of different algorithms are as shown in Figure 5, (b) and QDEMA algorithm is slightly different from the comparison algorithms according to the figure.



(a) Station-to-station Distance Diagram



(b) Distribution Route Comparison

Figure 5. Route Diagram of Distribution Scheme

6. Conclusion

According to the characteristics of the two-echelon vehicle routing optimization problem, this problem is decomposed into the first-echelon transfer station optimization which aims at adopting the optimal cut algorithm for accurate distribution and the second-echelon client optimization which aims at adopting Q learning theory and differential evolution algorithm to design new Memetic algorithm (QDEMA), thus to effectively decouple the two-echelon vehicle routing optimization problem. The simulation result shows: compared with the comparison algorithms, the proposed algorithm can more effectively solve the two-echelon routing optimization problem.

Acknowledgment

The research is supported by the following project:
(1) The National Natural Science Fund(No.61401101)
(2) The natural science foundation of Anhui Province (No.1408085QF122)

(3) Natural Science Foundation of Anhui Provincial Education Department (No.KJ2015A295, No.2015KJ014, No.2015KJ007)
(4) The National College Students' innovative projects (No.201410371030, No.AH201410371037)
(5) The national statistical science project plan research(No.2014LZ32)
(6) Fuyang Normal University Foundation(NO.2013ZYSD05,No.2013KJFH05, No.2013FSKJ14, No.2015KJFH02，No.2015FSKJ09)
(7) Anhui Province Quality Project(No.2014zdjy083, No.2014zdjy080,No.2014zy048)

Reference

[1] J. Hu and Z. Gao, “Distinction immune genes of hepatitis-induced heptatocellular carcinoma”, Bioinformatics, vol. 28, no. 24, **(2012)**, pp. 3191-3194.
[2] J. Yang, B. Chen and J. Zhou, “A Low-Power and Portable Biomedical Device for Respiratory Monitoring with a Stable Power Source”, Sensors, vol. 15, no. 8, **(2015)**, pp. 19618-19632.
[3] G. Bao, L. Mi, Y. Geng and K. Pahlavan, “A computer vision based speed estimation technique for localiz ing the wireless capsule endoscope inside small intestine”, 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), **(2014)** August.
[4] X. Song and Y. Geng, “Distributed community detection optimization algorithm for complex networks”, Journal of Networks, vol. 9, no. 10, **(2014)** January, pp. 2758-2765.
[5] D. Jiang, X. Ying and Y. Han, “Collaborative multi-hop routing in cognitive wireless networks”, Wireless Personal Communications, **(2015)**, pp. 1-23.
[6] Z. Lv, A. Halawani and S. Feng, “Multimodal hand and foot gesture interaction for handheld devices”, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 11, no. 1, **(2014)**, pp. 10.
[7] G. Liu, Y. Geng and K. Pahlavan, “Effects of calibration RFID tags on performance of inertial navigation in indoor environment”, 2015 International Conference on Computing, Networking and Communications (ICNC), **(2015)** Febuary.
[8] J. He, Y. Geng, Y. Wan, S. Li and K. Pahlavan, “A cyber physical test-bed for virtualization of RF access environment for body sensor network”, IEEE Sensor Journal, vol. 13, no. 10, **(2013)** October, pp. 3826-3836.
[9] J. He, Y. Geng and K. Pahlavan, “Toward Accurate Human Tracking: Modeling Time-of-Arrival for Wireless Wearable Sensors in Multipath Environment”, IEEE Sensor Journal, vol. 14, no. 11, **(2014)** November, pp. 3996-4006.
[10] Z. Lv, A. Halawani and S. Fen, “Touch-less Interactive Augmented Reality Game on Vision Based Wearable Device”, Personal and Ubiquitous Computing, vol. 19, no. 3, **(2015)**, pp. 551-567.
[11] G. Bao, L. Mi, Y. Geng, M. Zhou and K. Pahlavan, “A video-based speed estimation technique for localizing the wireless capsule endoscope inside gastrointestinal tract”, 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), **(2014)** August.
[12] D. Zeng and Y. Geng, “Content distribution mechanism in mobile P2P network”, Journal of Networks, vol. 9, no. 5, **(2014)** January, pp. 1229-1236.
[13] W. Gu, Z. Lv and M. Hao, “Change detection method for remote sensing images based on an improved Markov random field”, Multimedia Tools and Applications, **(2015)**, pp. 1-16.
[14] W. Huang and Y. Geng, “Identification Method of Attack Path Based on Immune Intrusion Detection”, Journal of Networks, vol. 9, no. 4, **(2014)** January, pp. 964-971.
[15] X. Li, Z. Lv and J. Hu, “XEarth: A 3D GIS Platform for managing massive city information”, Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2015 IEEE International Conference on. IEEE, **(2015)**, pp. 1-6.

Author

Liu Dongdong, received his M.S. degree in Computer application from Dalian Maritime University in Dalian, China. He is currently a lecturer in the Fuyang Normal University. His research interest is mainly in the area of Computer Software, Network, Algorithm optimization.

