Emergency logistics vehicle scheduling based on improved differential evolution

Yazhou Shan[1*], Kaijun Wu[1], Yuechun Wu[2]

[1] *School of Electronic and Information Engineering, LanZhou JiaoTong Unversity, Lanzhou 730070, China*

[2] *Department of Qingdao signal depot, Jinan Railway Bureau, Qingdao 266002, China*

Received 21 August 2014, www.cmnt.lv

Abstract

Emergency logistics vehicle scheduling is an issue of NP combination which possesses important practical value. In order to overcome the problems such as long computing time and easy to fall into local optimal solution for traditional heuristic optimization algorithm, an improved differential evolution is proposed. In the algorithm, a greedy algorithm is used to generate the initial population, legalized method is used to repair mutation and improved order crossover is used, then, a new selection mechanism is added in after the mutation operator. In the paper, the algorithm is used to solve emergency logistics vehicle scheduling problem, the mathematic mode which minimize total cost with the emergency time constraint is established and the solution algorithm is developed. The simulation results of example indicate that the algorithm can efficiently solve emergency logistics vehicle scheduling problem through the comparison of GA and the DE algorithm.

Keywords: emergency logistics vehicle scheduling; differential evolution; NP combination problem; greedy algorithm

1 Introduction

With the rapid development of social economy and increasing levels of urban modernization, all kinds of emergencies have enormous influences on urban functions. Some unexpected events will seriously affect the life safety of the people, such as earthquakes, tsunamis, gas leak. Losses stemmed from ineffective emergency logistics account for 15% to 20% of the entire casualty and financial losses caused by sudden outbursts of natural calamities and human-contrived disasters.

Emergency logistics vehicle scheduling is a special goods transporting activity which aims at providing emergency resource for emergencies, maximizing efficiency and minimizing damages caused by disasters. Emergency logistics vehicle scheduling is an important component in dealing with accidents, which in time weighs great value in reducing losses, preventing seconddary disasters and maintaining social stability.

When a natural disaster-stricken spot is in need of countermeasures from emergency resource transporting system, emergency vehicles should deliver emergency resource to demand spot within the least of time. Emergency logistics vehicle scheduling is a typical case of optimizing combination, a tough issue concerning NP. It contains great complexities. Questions of the similar kind are hot studied by different branches of science, such as operations research, applied mathematics, computer application, graph theory and communication & transportation, etc. The time it takes to be solved is growing exponentially as its dimension is expanding, which is hard to be handled with traditional mathematic methods. In recent years, heuristic optimization algorithms such as Genetic Algorithm, Ant Colony Algorithm and Particle Swarm Optimization have been widely applied to solving such kind of problems[1-3]; however, these algorithms all have drawbacks like long computing time, easy to fall into stagnation and unable to do further computing. Therefore, how to construct optimized algorithm simple in formation and high in optimizing precision enjoys great significance in solving emergency logistics vehicle scheduling problem.

Differential Evolution (DE) is a bionic intelligent algorithm, mimicking the natural evolutionary law Survival of the Fittest [4]. DE performs better in solving complicated global optimization, and is brief in process [5-6]. It has less controlled parameters and is regarded as a significant breakthrough in the aspect of algorithm structure since the invention of bionic intelligence algorithm [7]. To overcome its shortcomings of low convergence speed, huge computation and inclination to becoming premature, an improved Differential Evolution (IDE) is proposed in the thesis. The result of experiments indicates that the algorithm can efficiently solve Emergency logistics vehicle scheduling problem.

2 The Mathematic Mode of Emergency logistics vehicle scheduling Problem

2.1 PROBLEM DESCRIPTION

There are plenty of emergency vehicles in the emergency rescue system. The vehicles will set off from the rescue center and deliver goods to several crisis-attacked sites (quantity=R). After delivery, the vehicles will return to the center. The maximum capacity for each vehicle is P_k ($k = 1,2,..., K$). Find driving routes from the center to each demanding site meeting the following requirements [8]:

* ***Corresponding author's*** e-mail: 754123815@qq.com

1). Overload is forbidden.
2). Delivery must be finished as soon as possible.
3). Supplies required by each demanding site must be delivered.
4). The cost of emergency center must be at minimum.

Supplementary conditions:

1). Adequate vehicles are available in emergency center, and the capacity of each has been given in the discussion;
2). There is only one emergency center and its location has been given in the discussion;
3). There is only one type vehicle in the center. Each crisis-attacked site is served by only one vehicle. Every site must be covered by the driving route;
4). The gross demand of each demanding site along the route cannot surpass the capacity of the vehicle;
5). In order to simplify the case, only the time consumed on road will be considered, irrespective of the service time.

MATHEMATIC MODE

Here is the mathematic mode built according to the description above:

(1)---objective function (2)-(11)---constraint conditions

$$\min z = \sum_{k\in M} x_{k_0} c_k^f + \sum_{k\in M}\sum_{i\in s}\sum_{j\in s} c_{ijk}^t d_{ij} y_{ijk} \quad (1)$$

$$\sum_{i\in S}\sum_{k\in M} y_{ijk} = 1, \forall j \in N \quad (2)$$

$$\sum_{i\in s} y_{ihk} - \sum_{i\in s} y_{hik} = 0, \forall k \in M, \forall h \in S \quad (3)$$

$$\sum_{j\in N}\sum_{k\in M} y_{0jk} > 0 \quad (4)$$

$$\sum_{i\in s}\sum_{j\in N} y_{ijk} q_j \le Q_k, \forall k \in M \quad (5)$$

$$\sum_{k\in M}\sum_{i\in S}\sum_{j\in N} y_{ijk} q_j \le Q_0 \quad (6)$$

$$t_{jik} = d_{ij} / v_{ijk} \quad (7)$$

$$t_{ik} = t_{jk} + t_{jik} y_{jik}, \forall i, j \in S, \forall k \in M \quad (8)$$

$$t_{ik} \le l_i, \forall i \in N, \forall k \in M \quad (9)$$

$$y_{ijk} = \{0,1\}, \forall i, j \in S, \forall k \in M \quad (10)$$

$$x_{k0} = \{0,1\}, \forall k \in M \quad (11)$$

In the mode:

A. d_{ij} refers to the distance between node i and node j ($i, j = 0,1,2\cdots S,$). When $i, j = 0$, d_{ij} refers to the emergency center;

B. v_{ijk} refers to the average speed of vehicle k from node i to node j;

C. t_{ik} refers to the time vehicle k needs to go to the demanding site i;

D. t_{ijk} refers to the time vehicle k needs to go from node i to node j;

E. l_i refers to the latest time of arrival of vehicle k;

F. n_k refers to the total number of demanding sites that vehicle k needs to serve. When $n_k = 0$, it means that the vehicle does not participate in the mission.

G. R_k refers to the set of demanding sites served by the vehicle;

H. c_k^f refers to the fixed cost of vehicle k when driving;

I. c_{ijk}^t refers to the unit cost of vehicle k from node i to node j;

J. Q_k refers to the loading capacity of vehicle k;

K. q_i refers to the demand of node i;

L. N refers to the set of crisis-attacked sites. $N = \{r \mid r = 1,2,..., R\}$;

M. M refers to the set of emergency vehicles. $M = \{k \mid k = 1,2,..., K\}$;

N. S refers to the sum of nodes. $S = N \cup O$.

x_{k0} and y_{ijk} are defined as follows:

$$x_{k0} = \begin{cases} 1 & \text{emergency vehicle k is in use} \\ 0 & \text{otherwise} \end{cases};$$

$$y_{ijk} = \begin{cases} 1 & \text{vehicle k drive from node i to node j, and } i \ne j \\ 0 & \text{otherwise} \end{cases}$$

3 Algorithm to Solve Emergency logistics vehicle scheduling Problem Based on IDE

3.1 ENCODING METHOD

Brief and direct sequence coding is used. The set of crisis-attacked sites' sequence number is {1,2,…,n}. The sequence number of the parking lot is 0. The number of vehicles is k. Encoding structure is $\{v_1, v_2,..., v_i,..., v_{n+k-1}\}$.

It is a random combination of the set of crisis-attacked sites' sequence number {1,2,…,n}and 0 (quantity =k-1). The adding of 0(quantity=k-1) is to differentiate different driving route of vehicles.

DECODING METHOD

(1). Add 0 to both ends of each coding;
(2). Traverse each coding and save the non zero sequence that has been separated by zero in the coding, which is the driving route of the vehicle.

That is to say, 1 to k driving route(s) can be decoded from each coding.

For example, there are 9 crisis-attacked sites and 5 vehicles. Then the coding will be

X=(0,1,2,3,0,0,4,5,6,0,7,8,9).

The decoding will be 3 sets of non zero sequences (1,2,3), (4,5,6), (7,8,9). This coding can be interpreted as:

three vehicles have been put into use and they follow the following driving routes respectively

(1→2→3), (4→5→6), (7→8→9).

No negative coding, decimal coding and repetition of non zero bits are permitted in the coding. That is the requirement of legalization and it is also the reason why legalization is needed after mutation operation.

3.2 FITNESS FUNCTION

Define fitness function according to mode1:

It is the weighted sum of the number of used vehicles Z_1 and the total length of all vehicles' driving routes Z_2 .

The dereferencing of weight coefficient k_1 and k_2 is set based on practice. If priority is given to less number of vehicles, then $k_1 \gg k_2$ can be set; if priority is given to shorter distance of driving route, then $k_1 \ll k_2$ can be set.

3.3 POPULATION INITIALIZATION

In order to improve the quality of initial population, greedy initializing method is used. Generating method of each individual in the initial population is as following:

step1: The list of to be initialized crisis-attacked sites is List, containing all the sequence of crisis-attacked sites. SList is the list of initialized crisis-attacked sites and it is null;

step2: Select siteA at random as the start point, and make it current crisis-attacked siteT. Insert siteT to SList and then remove it from List ;

step3: Pick a site nearest to siteT as the new current crisis-attacked siteT from List. Insert T to SList and remove from List;

step4: Estimate if List is null. If so, turn to step5; otherwise, turn to step3;

step5: Insert 0(quantity=(K-1)) to SList randomly;

step6: Estimate if SList meets all the constraint conditions. If not, turn to step1; if so, end the initialization.

By the greedy initializing method above to initialize population, the local optimal solution can be achieved and convergence speed can be improved by a large margin.

3.4 PROCESS OF MUTATION AND LEGALIZATION

Adding and subtracting the randomly picked individual bitwise, and then legalize it.

For example, there are 5 sites and 2 vehicles,. Pick 3 individuals at random:

(1,0,2,3,0,4,5)、(2,5,0,3,1,0,4)、(5,3,1,0,0,4,2).

If F=1, $v_i = x_{r_1} + F \bullet (x_{r_2} - x_{r_3})$ is applied to each individual. The consequence is v_i=(-2,2,1,6,1,0,7). In this case, there are negative coding and repetition of non zero bits in v_i, which is illegal coding needing legalization.

The process of legalization:

(1) Detect 0 in coding v_i and record their locations. If the number of 0 surpasses that of vehicles, then detect the same quantity of 0 as the vehicle. v_i=(-2,2,1,6,1,7)

(2) Sort out coding v_i that has been detected from the smallest to the biggest. Replace its original coding value with its sequence number.

For example, v_i=(-2,2,1,6,1,7),

after sorting: (-2,1,1,2,6,7).

Replace its original coding value with its sequence number :(1, 4, 2, 5, 3, 6)

(3) Inset 0 that has been detected in step1 to coding, then v_i = (1, 4, 2, 5, 3, 0, 6)

3.5 IMPROVED GREEDY ORDER CROSSOVER

Order crossover can be seen as the deformation of partially mapped crossover PMX with different fixes. It performs better in retaining the adjacent relationship and sequencing. Improvements mainly focus on the possibility of repetition of the encoding value of 0 bits. That is to say, when the given gene is removed, all the 0 bits in the gene must be deleted from the beginning to the end,

The steps are as follows:

Step1: Choose two tangent point's c1and c2 ;

Step2: Exchange the middle part ;

Step3: List the original order from the first gene behind the second tangent point c2, and remove the given gene. Pay attention to the quantity of 0.

Step4: Fill the achieved non duplicate order in the blank from the first seat behind the second tangent point c2, schematic diagram of the crossover operation is shown in figure 1.

Each time of order crossover will produce two cross individuals while DE crossover operator needs only one cross individual. Therefore, in order to improve the convergence speed, fitness individuals as a return to the cross individual, improved algorithm chooses better adapted individuals as the return crossover individuals using the greedy selection mechanism.

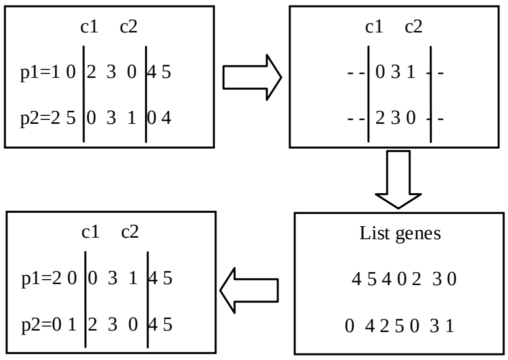


Figure 1 diagram of Crossover

3.6 OPERATIONS OF CHOOSING AND IMPROVEMENT IN TECHNICAL PROCESS

After the mutation operation of the basic differential evolution algorithm, jumping to crossover operation directly may lead to destruction of good genes produced in mutation operation. In order to prevent cross operation to eliminate or affect the optimization effect produced in mutation operation, apart from retaining the greedy selection mechanism, the improved algorithm retains the selection mechanism after mutation operation. That is , if the mutation individual adapts better than the original individual, skip the crossover operation and choose individual mutation to put into the next generation of the population; if the individual mutation is inferior to the original individual, then conduct further crossover operation based on mutation according to the process of the basic differential evolution algorithm

3.7 THE WHOLE PROCESS OF IDE

step1: Initialize parameter and let iteration number t=0. Set the maximum number of iterations, population size NP, scaling factor F, and the crossover constant CR.
step2: Initialize the greedy selection mechanism
step3: Frequency of cycling t←t+1.
step4: Set target individual index number i=1.
step5: Choose three different individuals at random apart from the target individual x_i.
step6: Conduct mutation and produce mutated individual v_i.
step7: Calculate the adaptation of v_i and perform the selecting operation. If the mutated individual v_i is better than the target individual x_i, then turn to step10; otherwise, turn to step8.
step8: Perform crossover operation to x_i and v_i, produce crossover individual u_i.
step9: Calculate the adaptation of u_i and perform the selecting operation.
step10: Target individual index number i←i+1, return to step5 and perform it repeatedly until i=NP; otherwise, perform step11.
step11: If the number of iterations is bigger than the maximum number of iterations, then end the cycle and output the calculation results; otherwise jump to the step3 and continue the next iteration.

Turn to step2.

Where k_i is the number of nodes that are cluster head in each layer and N_i is the total nodes number in each layer.

4 Analysis of the simulation results of example

4.1 EXAMPLE 1

A crisis-attacked area has 8 demand sites needing resource delivered from emergency resource center. There are 5 vehicles. The loading capacity is 8 ton for each vehicle. The fixed cost of each is 80 Yuan. The average speed is 20km/hour. The average driving cost is 10yuan/km. the distance between sites, the demand (unit: ton)and the latest arrival time (unit: minute) are displayed in tab1(0 represents for the center, 1-8 represents for the sites).

TABLE1 The data of distribution centre and disaster location

Crisis-attacked sites	Crisis-attacked sites								
	0	1	2	3	4	5	6	7	8
0	0	4	6	7.5	9	20	10	16	8
1	4	0	6.5	4	10	5	7.5	11	10
2	6	6.5	0	7.5	10	10	7.5	7.5	7.5
3	7.5	4	7.5	0	10	5	9	9	15
4	9	10	10	10	0	10	7.5	7.5	10
5	20	5	10	5	10	0	7	9	7.5
6	10	7.5	7.5	9	7.5	7	0	7	10
7	15	11	7.5	9	7.5	9	7	0	10
8	8	10	7.5	15	10	7.5	10	10	0
q_i		3	3	1	3	2	4	1	4
l_i		40	60	30	80	80	50	90	30

Compute the example with GA, DE and IDE respecttively on the same computer. The optimal scheduling plan is that 3 cars participate in the task. The driving route, distance and cost are shown in tab 2. Objective function changes according to iteration and its tendency is recorded in figure 2.

TABLE 2 The optimal scheduling scheme of example 1

vehicles	driving routes	driving time	driving time	total cost
vehicle1	0—1—3—2—0	64.5	21.5	295
vehicle2	0—6—4—0	79.5	26.5	345
vehicle3	0—8—5—7—0	118.5	39.5	475
total		262.5	87.5	1115

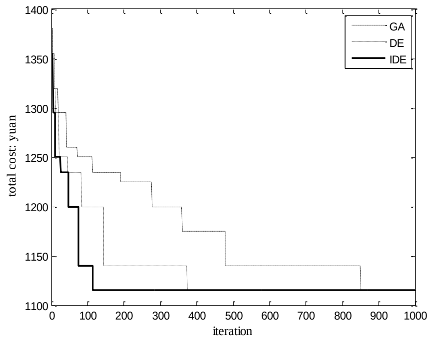


Figure 2 Performance comparison of GA, DE and IDE

4.2 EXAMPLE 2

A crisis-attacked area has 20 demand sites needing resource delivered from emergency resource center. There are 4 vehicles. The loading capacity is 20 ton for each vehicle. The fixed cost of each is 100 yuan. The average speed is 25km/hour. The average driving cost is 12 yuan/km. The coordinate of emergency rescue center is 0 point (3km, 4km), position coordinates of demand sites (unit: km), the demand q_i (unit: ton)and the latest arrival time l_i (unit: minute) $(i=1,2,...,20)$ are displayed in tab1(0 represents for the center, 1-20 represents for the sites).

TABLE 3 The data of emergency center and emergency location

order number	x coordinates	y coordinates	demand	latest arrival time
0	3	4		
1	3.1	7.6	4	60
2	1	8.6	2	15
3	6.3	5.7	1	60
4	2	3.3	1	45
5	5.3	6.6	5	10
6	6	9.8	4	60
7	0.6	9.4	2	30
8	3	4.5	2	10
9	1.9	9.8	3	20
10	3.7	7.1	2	45
11	8.4	8	2	30
12	4.7	9.3	1	30
13	1.4	6.8	2	15
14	0.3	6.3	2	30
15	3.5	0.5	3	60
16	2.4	1.6	4	50
17	8.2	3.7	4	45
18	7.3	8.4	3	60
19	8.5	5.2	1	50
20	7.8	8.8	3	20

Compute the example with GA, DE and IDE respecttively on the same computer. The optimal scheduling plan is that 3 cars participate in the task. The driving route, distance and cost are shown in tab 4. Objective function changes according to iteration and its tendency is recorded in figure 3.

TABLE 4 The optimal scheduling scheme of example 2

vehicles	driving routes	driving time	driving time	total cost
vehicle1	0—5—20—11—19—17—18—3—0	56.42	23.51	382.12
vehicle2	0—8—13—14—4—16—15—0	47.55	14.8	277.55
vehicle3	0—2—7—9—12—6—10—1—0	46.65	19.44	333.24
total		150.62	57.74	992.91

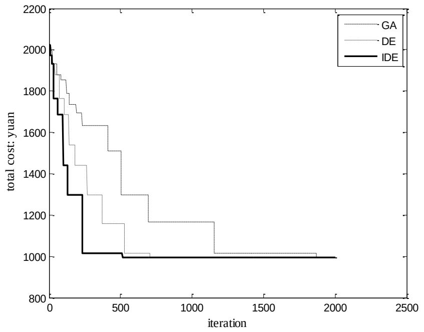


Figure 3 Performance comparison of GA, DE and IDE

It is obvious from the result of the two experiments that the algorithm proposed in the thesis can detect the optimal solution to emergency logistics vehicle scheduling problem swiftly and accurately. Its efficiency is better than that of GA and DE. It is a new method to solve the emergency logistics vehicle scheduling problem.

5 Conclusions

To solve the problem of NP in the emergency logistics vehicle scheduling, an improved method that differential evolution algorithm has been applied to the emergency logistics vehicle scheduling is proposed in the thesis. The simulation results of example indicate that the algorithm can efficiently solve emergency logistics vehicle scheduling problem through the comparison of GA and the DE algorithm.

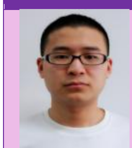
Acknowledgments

This work was supported by the by the National Social Science foundation of China**(**No.12CGL004**)**, the Science and Technology Support Project of Gansu Province (No.1304FKCA097), the Natural Science Fund Project of Gansu Province (No. No.1212RJZA063 and No.1310RJZA039) and the Scientific Research Project of Colleges and Universities in Gansu Province (No.2013A-052).

References

[1] Ren J, Xi H W , Shi X F. Genetic Algorithm for Vehicle Scheduling Problem of City Emergency Logistics Distribution [J], Journal of Millitary Transportation University, 13(9) , 2011, 70-73.
[2] Tang L S, Cheng W M, Liang J, et al. Research on Ant Colony Clustering Algorithm for the Problem of Emergency Logistics Distribution [J], Railway Transport and Economy, 30(9), 2008,66-69.
[3] Wu K J, Wang T J. The model and optimization algorithm of multi-depot emergency materials dispatching with time limits [J],Computer Engineering and Applications, 48(30), 2012, 19-23.
[4] Storn P, Price K. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Berkley: International Computer Science Institute, 1995.
[5] Vesterstorm J, Thomsen R. A comparative study of differential evolution, particle swarm optimization and evolutionary algorithm on numerical benchmark problem. Proceedings of IEEE Congress on Evolutionary Computation, 2004:1980-1987.
[6] Duan H B, Zhang X Y, Xu C F. Bionic Intelligent Computing, Science Press, Beijing, 2011
[7] Bo Liu, Ling Wang, Jin Y H. Advances in differential evolution [J], Control and Decision, 22(7), 2007,721-729
[8] Guo H L. Study on Multi-Objective Vehicle Routing Problem in urban Emergency Logistics, Baoding ,Hebei University ,2010
[9] Wang D W. Intelligent optimization method, Higher Education Press, Beijing, 2007

Authors



Yazhou Shan, born in Henan, P.R. China, June 27, 1990

Current position, grades: master student in the lanzhou jiaotong university
University studies: master degree will be received in the school of Electronic and Information Engineering, LanZhou JiaoTong in 2016.
scientific interests: Vehicle routing problem, intelligent optimization algorithm, etc.

Kaijun Wu, born in Shandong, P.R. China, March 1, 1978

Current position, grades: associate professor and master tutor in the lanzhou jiaotong university
University studies: PhD student of automation and electrical engineering, lanzhou jiaotong university, P.R.
scientific interests: Emergency logistics scheduling, nonlinear dynamics, etc.
Publications: 35 papers,2 textbooks.

Yuechun Wu, born in Shandong, P.R. China, May 23, 1987

Current position, grades: Assistant engineer in the department of Qingdao signal depot, Jinan Railway Bureau
University studies: Master graduate in the school of automation and electrical engineering, lanzhou jiaotong university, P.R.
scientific interests: Traffic flow forecasting, neural network ,etc.
Publications: 6 papers.