



Journal of  
**Software  
Engineering**

ISSN 1819-4311



Academic  
Journals Inc.

[www.academicjournals.com](http://www.academicjournals.com)

## **Emergency Logistics Distribution VRP Based on Differential Evolution Optimization**

<sup>1</sup>Pei Tang and <sup>2</sup>Shanhong Zhu

<sup>1</sup>Xinxiang University, Henan, China

<sup>2</sup>International School of Software, Wuhan University, Wuhan, China

*Corresponding Author: Pei Tang, Xinxiang University, Henan, China*

### **ABSTRACT**

Combined with emergency logistics distribution vehicle routing problem, in the light of the outstanding characteristic of emergency logistics, an improved differential evolution is proposed. The mathematic mode which minimize total cost with the emergency time constraint is established and the solution algorithm is developed and the greedy algorithm optimization is used to optimize the distribution path. Results of example's analysis show that this proposed model, algorithm has rationality and feasibility.

**Key words:** Emergency logistics, VRP, differential evolution, greedy algorithm

### **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 and tsunamis, gas leak. Losses stemmed from ineffective emergency logistics account for 15-20% 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 secondary 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 and 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 (Ren *et al.*, 2011; Tang *et al.*, 2008; Wu and Wang, 2012); 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 is a bionic intelligent algorithm, mimicking the natural evolutionary law Survival of the Fittest (Storn and Price, 1995). DE performs better in solving complicated global optimization and is brief in process (Vesterstrom and Thomsen, 2004; Duan *et al.*, 2011). 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 (Liu *et al.*, 2007). To overcome its shortcomings of low convergence speed, huge computation and inclination to becoming premature, an Improved Differential Evolution (IDE) is proposed in the study. The result of experiments indicates that the algorithm can efficiently solve emergency logistics vehicle scheduling problem.

## MATHEMATIC MODE OF EMERGENCY LOGISTICS VEHICLE SCHEDULING PROBLEM

**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  $P_k$  ( $k = 1, 2, \dots, K$ ). Finding driving routes from the center to each demanding site, meet the following requirements (Guo, 2010):

- Overload is forbidden
- Delivery must be finished as soon as possible
- Supplies required by each demanding site must be delivered
- The cost of emergency center must be at minimum

### Supplementary conditions:

- Adequate vehicles are available in emergency center and the capacity of each has been given in the discussion
- There is only one emergency center and its location has been given in the discussion
- 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
- The gross demand of each demanding site along the route cannot surpass the capacity of the vehicle
- 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. Objective function is described in Eq. 1:

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

Constraint conditions are described in Eq. 2-11:

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

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

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

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

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

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

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

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

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

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

Where:

$d_{ij}$  = Distance between node I and node j ( $i, j = 0, 1, 2, \dots, S$ ). When  $i, j = 0$ ,  $d_{ij}$  is to the emergency center

$v_{ijk}$  = Average speed of vehicle k from node i to node j

$t_{ik}$  = Time vehicle k needs to go to the demanding site i

$t_{ijk}$  = Time vehicle k needs to go from node i to node j

$l_i$  = Latest time of arrival of vehicle k

$n_k$  = 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

$R_k$  = Set of demanding sites served by the vehicle

$c_k^f$  = Fixed cost of vehicle k when driving

$c_{ijk}^t$  = Unit cost of vehicle k from node i to node j

$Q_k$  = Loading capacity of vehicle k

$q_i$  = Demand of node i

$N$  = Set of crisis-attacked sites.  $N = \{r \mid r = 1, 2, \dots, R\}$

$M$  = Set of emergency vehicles.  $M = \{k \mid k = 1, 2, \dots, K\}$

$S$  = Sum of nodes.  $S = N \cup O$

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

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

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

## ALGORITHM TO SOLVE EMERGENCY LOGISTICS VEHICLE SCHEDULING PROBLEM BASED ON IDE

**Encoding method:** Brief and direct sequence coding is used in encoding method. The set of crisis-attacked sites' sequence number is  $\{1, 2, \dots, n\}$ . The sequence number of the parking lot is 0. The number of vehicles is  $k$ . Encoding structure is  $\{v_1, v_2, \dots, v_i, \dots, v_{n+k-1}\}$ . It is a random combination of the set of crisis-attacked sites' sequence number  $\{1, 2, \dots, n\}$  and 0 (quantity =  $k-1$ ). The adding of 0 (quantity =  $k-1$ ) is to differentiate different driving route of vehicles.

### Decoding method:

- Add 0 to both ends of each coding
- 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 routes 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.

**Fitness function:** Fitness function is defined according to model1:

- 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  $k_2 \gg k_1$  can be set; if priority is given to shorter distance of driving route, then  $k_1 \gg k_2$  can be set

**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:

- Step 1:** 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
- Step 2:** 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
- Step 3:** Pick a site nearest to siteT as the new current crisis-attacked siteT from List. Insert T to SList and remove from list
- Step 4:** Estimate if list is null. If so, turn to step 5; otherwise, turn to step 3
- Step 5:** Insert 0(quantity =  $(K-1)$ ) to SList randomly
- Step 6:** Estimate if SList meets all the constraint conditions. If not, turn to step 1; 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.

**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  $F = 1$ ,  $v_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3})$  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 is as follows:

- 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.  $v_i = (-2, 2, 1, 6, 1, 7)$
- 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)
- Inset 0 that has been detected in step1 to coding, then  $v_i = (1, 4, 2, 5, 3, 0, 6)$

**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:

**Step 1:** Choose two tangent points  $c1$  and  $c2$

**Step 2:** Exchange the middle part

**Step 3:** 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

**Step 4:** 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 Fig. 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.

**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

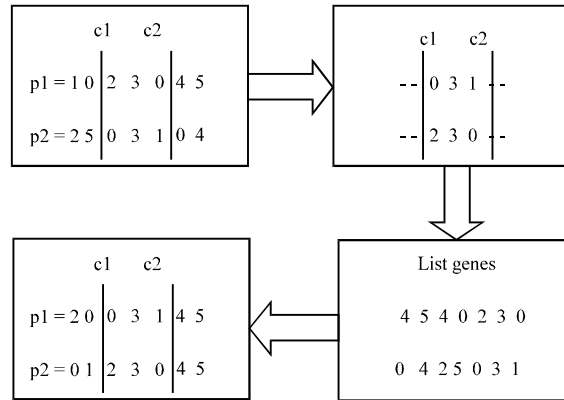


Fig. 1: Diagram of crossover

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.

**Whole process of IDE:**

- Step 1:** 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
- Step 2:** Initialize the greedy selection mechanism
- Step 3:** Frequency of cycling  $t \leftarrow t+1$
- Step 4:** Set target individual index number  $i = 1$
- Step 5:** Choose three different individuals at random apart from the target individual  $x_i$
- Step 6:** Conduct mutation and produce mutated individual  $v_i$
- Step 7:** Calculate the adaption 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 step 10; otherwise, turn to step 8
- Step 8:** Perform crossover operation to  $x_i$  and  $v_i$ , produce crossover individual  $u_i$
- Step 9:** Calculate the adaption of  $u_i$  and perform the selecting operation
- Step 10:** Target individual index number  $i \leftarrow i+1$ , return to step 5 and perform it repeatedly until  $i = NP$ ; otherwise, perform step 11
- Step 11:** 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 step 3 and continue the next iteration

Turn to step2.

**ANALYSIS OF THE SIMULATION RESULTS OF EXAMPLE**

**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 t for each vehicle. The fixed cost of

each is 80 yuan. The average speed is 20 km/h. The average driving cost is 10 yuan km<sup>-1</sup>. The distance between sites, the demand  $q_i$  (unit: ton) and the latest arrival time  $l_i$  (unit: Minute) ( $i = 1, 2, \dots, 8$ ) are displayed in Table 1 (0 is center, 1-8 is sites).

Compute the example with GA, DE and IDE, respectively 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 Table 2. Objective function changes according to iteration and its tendency is recorded in Fig. 2.

**Example 2:** A crisis-attacked area has 20 demand sites needing resource delivered from emergency resource center (Table 3). There are 4 vehicles. The loading capacity is 20 t for each vehicle.

Table 1: Data of distribution center and disaster location

Crisis-attacked sites	Crisis-attacked sites								
	0	1	2	3	4	5	6	7	8
0	0.0	4.0	6.0	7.5	9.0	20.0	10.0	16.0	8.0
1	4.0	0.0	6.5	4.0	10.0	5.0	7.5	11.0	10.0
2	6.0	6.5	0.0	7.5	10.0	10.0	7.5	7.5	7.5
3	7.5	4.0	7.5	0.0	10.0	5.0	9.0	9.0	15.0
4	9.0	10.0	10.0	10.0	0.0	10.0	7.5	7.5	10.0
5	20.0	5.0	10.0	5.0	10.0	0.0	7.0	9.0	7.5
6	10.0	7.5	7.5	9.0	7.5	7.0	0.0	7.0	10.0
7	15.0	11.0	7.5	9.0	7.5	9.0	7.0	0.0	10.0
8	8.0	10.0	7.5	15.0	10.0	7.5	10.0	10.0	0.0
$q_i$		3.0	3.0	1.0	3.0	2.0	4.0	1.0	4.0
$l_i$		40.0	60.0	30.0	80.0	80.0	50.0	90.0	30.0

Table 2: Optimal scheduling scheme of example 1

Vehicles	Driving routes	Driving time	Driving time	Total cost
1	0-1-3-2-0	64.5	21.5	295
2	0-6-4-0	79.5	26.5	345
3	0-8-5-7-0	118.5	39.5	475
Total		262.5	87.5	1115

Table 3: Data of emergency center and emergency location

Order				Latest	Order				Latest
No.	x coordinates	y coordinates	Demand	arrival time	No.	x coordinates	y coordinates	Demand	arrival time
0	3.0	4.0			11	8.4	8.0	2	30
1	3.1	7.6	4	60	12	4.7	9.3	1	30
2	1.0	8.6	2	15	13	1.4	6.8	2	15
3	6.3	5.7	1	60	14	0.3	6.3	2	30
4	2.0	3.3	1	45	15	3.5	0.5	3	60
5	5.3	6.6	5	10	16	2.4	1.6	4	50
6	6.0	9.8	4	60	17	8.2	3.7	4	45
7	0.6	9.4	2	30	18	7.3	8.4	3	60
8	3.0	4.5	2	10	19	8.5	5.2	1	50
9	1.9	9.8	3	20	20	7.8	8.8	3	20
10	3.7	7.1	2	45					



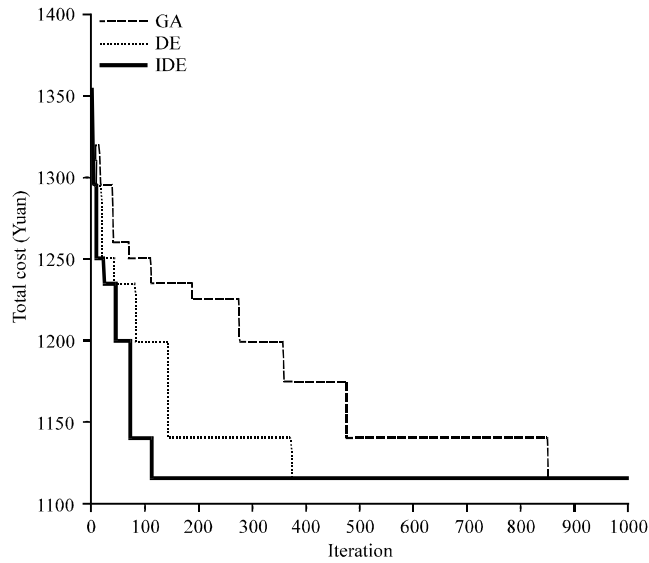


Fig. 2: Performance comparison of GA, DE and IDE

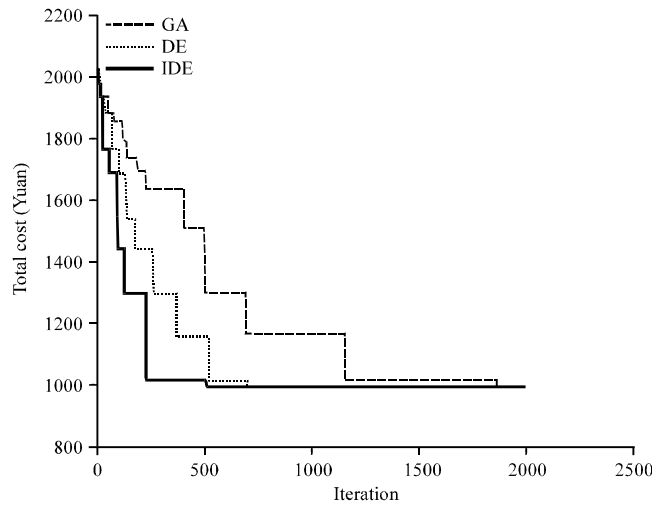


Fig. 3: Performance comparison of GA, DE and IDE

The fixed cost of each is 100 yuan. The average speed is 25 km h<sup>-1</sup>. The average driving cost is 12 yuan km<sup>-1</sup>. The coordinate of emergency rescue center is 0 point (3 and 4 km), position coordinates of demand sites (unit: km), the demand  $q_i$  (unit: t) and the latest arrival time  $l_i$  (unit: minute) ( $i = 1, 2, \dots, 20$ ) are displayed in Table 1 (0 is the center, 1-20 is the sites).

Compute the example with GA, DE and IDE, respectively 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 Table 4. Objective function changes according to iteration and its tendency is recorded in Fig. 3.

Table 4: Optimal scheduling scheme of example 2

Vehicles	Driving routes	Driving time	Driving time	Total cost
1	0-5-20-11-19-17-18-3-0	56.42	23.51	382.12
2	0-8-13-14-4-16-15-0	47.55	14.80	277.55
3	0-2-7-9-12-6-10-1-0	46.65	19.44	333.24
Total		150.62	57.74	992.91

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.

## CONCLUSION

In this study, an improved method that differential evolution algorithm has been applied to solve the problem of emergency logistics vehicle scheduling. The simulation results of example indicate that the algorithm can efficiently solve emergency logistics vehicle scheduling problem, it also can effectively solve the logistics distribution vehicle scheduling problem, it can shorten the transportation distance and transportation time, make the disaster losses to a minimum.

## ACKNOWLEDGMENT

This study is supported by the nature and science fund from Henan province ministry of education (Project No. JG [2013]13A520840).

## REFERENCES

- Duan, H.B., X.Y. Zhang and C.F. Xu, 2011. *Bionic Intelligent Computing*. Science Press, Beijing.
- Guo, H.L., 2010. Study on multi-objective vehicle routing problem in urban emergency logistics. Hebei University, Baoding.
- Liu, B., L. Wang and Y.H. Jin, 2007. Advances in differential evolution. *Control Decis.*, 22: 721-729.
- Ren, J., H. Xi and X. Shi, 2011. Genetic algorithm for vehicle scheduling problem of city emergency logistics distribution. *J. Military Transp. Univ.*, 13: 70-73.
- Storn, R. and K.V. Price, 1995. Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report, International Computer Science Institute, Berkley, CA., USA.
- Tang, L.S., W.M. Cheng, J. Liang and Z.Q. Zhang, 2008. Research on ant colony clustering algorithm for the problem of emergency logistics distribution. *Railway Transport Econ.*, 30: 66-69.
- Vesterstrom, J. and R. Thomsen, 2004. A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems. *Proceedings of the Congress on Evolutionary Computation*, Volume 2, June 19-23, 2004, Portland, OR., USA., pp: 1980-1987.
- Wu, K. and T. Wang, 2012. The model and optimization algorithm of multi-depot emergency materials dispatching with time limits. *Comput. Eng. Appl.*, 48: 19-23.