An Improved Differential Evolution Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pick-up Service

Erbao Cao [1,2], Mingyong Lai [1,2]

[1] *College of Economics and Trade, Hunan University, Changsha, 410079,China*

[2] *Hunan Province Laboratory of Logistics Information and Simulation Technology, Changsha. 410079,China*

Czp9491@163.com.cn

Abstract

The vehicle routing problem with simultaneous delivery and pick-up (VRP-SDP) is a variant of the classical vehicle routing problem (VRP) where clients require simultaneous delivery and pick-up. Deliveries are supplied from a single depot at the beginning of the vehicle's service, while pick-up loads are taken to the same depot at the conclusion of the service. One important characteristic of this problem is that a vehicle's load in any given route is a mix of delivery and pick-up loads, at the same time in any route the vehicle can not violate some constraints ,for example the vehicle capacity and traveling distance constraints.

In this paper, VRP-SDP is presented from the point of strategic view that combined the logistics and reverse logistics (bidirectional logistics). We constructed a universal mixed integer programming mathematic model of VRP-SDP in detail, it can transform into other classical vehicle routing problems by setting different parameters. An improved differential evolution algorithm (IDE) is proposed. In operation process, we firstly adopted the novel decimal coding to construct initial population, and then some improved differential evolution operators were adopted as the main optimizing scheme, such as adopted a real number coding method based on integer order criterion in mutation operation, a punishment function was designed to dispose constraints, and in crossover operation the crossover probability was self-updated with iteration. The computer simulations are used to compare the performance of the proposed method with genetic algorithm (GA), numerical results show that the performance of the proposed method is better than GA.

Keywords: Reverse logistics, improved differential evolution (IDE), integer programming, optimization

1. INTRODUCTION

Reverse logistics can be defined as the reverse process of logistics. The Council of Logistics Management (CLM) defines reverse logistics as "The process of planning, implementing, and controlling the efficient, cost effective flow of raw materials, in-process inventory, finished goods and related information from the point of consumption to the point of origin for the purpose of recapturing value or proper disposal" [1].In our work, VRP-SDP as an extension for vehicle routing problem, which is a complex combinational optimization problem, and is a well-know non-polynomial hard (NP-hard) Problem. VRP-SDP often encountered in fact, and has broad prospects in theory and practice, for example in the soft drink industry, where empty bottles must be returned, and in the delivery to grocery stores, where reusable pallets/containers are used for the transportation of merchandise. Reverse logistics is an important area in which the planning of vehicle routes takes the form of a VRP-SDP problem, as companies become interested in gaining control over the whole lifecycle of their products. For example, in some countries legislation forces companies to take responsibility for their products during lifetime, especially when environmental issue are involved(as in the disposal of laser printers' cartridges).Returned goods are another example where the definition of vehicle routes may take the form of a VRP-SDP problem. Owing to difficulty of the problem itself and deficiency of attention, even now little work can be found.

VRP-SDP is firstly proposed by Min H.[2], subsequently near 10 years, there are not correlative report until attach importance to reverse logistics, some researcher engaged in the problem[3]-[6]. Most of the algorithms of solving the VRP-SDP are based on that of classical VRP. In recent years, most published research for the VRP-SDP has focused on the development of heuristics. Genetic algorithm (GA) is a powerful algorithm for solving engineering design and optimization problems [7]-[9], and has been used to tackle many combinatorial problems, including certain types of vehicle routing problem. Storn and Price [10] first introduced the DE algorithm in 1996. DE was successfully applied to the optimization of some well-known nonlinear, non-differentiable and non-convex functions in Storn. DE combines simple arithmetic operators with the classical operators of crossover, mutation and selection to evolve from a randomly generated starting population to a final

0-7695-2875-9/07 $25.00 © 2007 IEEE

solution. DE is a population based and direct stochastic search algorithm (minimizer or maximizer), this simple, yet powerful and straightforward, features make it very attractive for numerical optimization. DE uses a rather greedy and less stochastic approach to problem solving compared to evolution algorithms. Recently, differential evolution algorithm have drawn great attention from researchers due to its robustness and flexibility and have been used to tackle many combinatorial problems, and its used field is fast expanding. But there are little work can be found about VRP that using differential evolution. In this paper, we developed a mixed integer programming mathematical model for VRP-SDP and proposed an improved differential evolution algorithm for the problem.

This paper is organized as follows: In section 2, we describe the vehicle routing problem with simultaneous delivery and pick-up service and present an integer programming mathematic model of VRP-SDP. In section 3, we design an improved differential evolution algorithm (IDE) to solve this model. Then we will give a numerical experiment to reveal the performance of the improved differential evolution algorithm in section 4. In section 5, we draw a conclusion.

2 . FORMULATION FOR VRP-SDP

There are $\bar{k}$ vehicles in the depot 0, and V stands for the set of customers to be visited, where n=$|V|$ is the number of customers. The location of depot and customers are known. Each customer has a known delivery demand level d_j and a know pick-up demand level p_j , j=1,2,…,n. Delivery routes for vehicles are required to start and finish at the depot, so that all customer demands are satisfied and each customer is visited just by one vehicle. $V_0 = V \cup \{0\}$ is the set of clients plus depot (client 0); c_{ij} is the distance between i and j, and the capacity of each vehicle is Q; the decision variable $x_{ijk} = 1$, if arc (i, j) belongs to the route operated by vehicle k, otherwise is 0 ; y_{ij} is the demand picked-up in clients routed up to node i and transported in arc (i, j); z_{ij} is the demand to be delivered to clients routed a after node i and transported in arc (i, j).

The corresponding integer programming mathematical formulation of VRP-SDP is given by:

$$\text{minimize} \quad \sum_{k=1}^{\bar{k}} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ijk} \qquad (1)$$

$$\text{s. t.} \quad \sum_{i=0}^{n} \sum_{k=1}^{\bar{k}} x_{ijk} = 1, \quad j=1,2,\ldots,n; \qquad (2)$$

$$\sum_{i=0}^{n} x_{ijk} - \sum_{i=0}^{n} x_{jik} = 0, \quad j=0,1,\ldots,n; k=0,1,\ldots,\bar{k}; \qquad (3)$$

$$\sum_{j=1}^{n} x_{0jk} \le 1, \quad k=1,2,\ldots,\bar{k}; \qquad (4)$$

$$\sum_{i=0}^{n} y_{ji} - \sum_{i=0}^{n} y_{ij} = p_j, \quad \forall j \ne 0; \qquad (5)$$

$$\sum_{i=0}^{n} z_{ij} - \sum_{i=0}^{n} z_{ji} = d_j, \quad \forall j \ne 0; \qquad (6)$$

$$y_{ij} + z_{ij} \le Q \sum_{k=1}^{\bar{k}} x_{ijk}, \quad i, j=0,1,\ldots,n; \qquad (7)$$

$$\sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} x_{ijk} \le L \quad k = 0,1,2,\ldots,\bar{k} \qquad (8)$$

$$x_{ijk} \in \{0,1\}, \ y_{ij} \ge 0, \ z_{ij} \ge 0, \ i, j=0,1,\ldots,n; \ k=0,1,\ldots,\bar{k}; \qquad (9)$$

The objective function seeks to minimize total distance traveled. Constraints (2) ensure that each client is visited by exactly one vehicle; constraints (3) guarantee that the same vehicle arrives and departs from each client it serves; Restrictions (4) define that at most $\bar{k}$ vehicles are used; restriction (5) and (6) are flow equations for pick-up and delivery demands, respectively; Constraints (7) establish that pick-up and delivery demands will only be transported using arcs included in the solution; Restrictions(8)are the maximum distance constraints, L is the upper limit on the total load transported by a vehicle in any given section of the route; Finally, constraints (9) define the nature of the decision variable。

The above formulation is very universal, and can easily turn into other classical vehicle routing problems. If p_j =0, then it transformed into conventional VRP formulation; If in someone customer, the all former clients p_j =0, and the followed clients d_j =0, then changes into VRP-B equivalent; If for all clients only have delivery or pick-up demand (either p_j or d_j equals 0), then it changes into VRP-PD equivalent; If only one vehicle can finish service, then it turns into TSP equivalent.

3. THE PROPOSED IDE FOR VRP-SDP

3.1 .Coding and fitness function

Like conventional GA for the VRP, a chromosome I(n) simply is a sequence (permutation) S of n customer nodes. We check the capacity constraints and distance constraints at the same time from the first gene of chromosome, if do not violate the constraints, considering the next gene; if it violate the constraints in someone gene, we consider to use other vehicle from this gene starting, and repeat the above process, till all of customers were serviced. That is to say the total amount of demands in a route can not exceed the capacity of the associated vehicle, and the total traveling distance can not over the maximum distance, For instance, there are 10 customer nodes, a randomly generated chromosome is 1 3 6 8 9 5 4 10 2 7,1 3 6 8 9 5 4 10 2 7, can be interpreted as r=3 feasible routes: 0-1-3-6-0, 0-8-9-5-0, and 0-4-10-2-7-0. If $\overline{k}$ >=r, then this chromosome is legal; otherwise, it is illegal.

In order to prevent illegal chromosome entering the next generation in great probability, a penalty function is designed. R is the total distance vehicles traveled of the corresponding chromosome, let $m = r - \overline{k}$, if $r > \overline{k}$, then $m > 0$, and $R = R + M \times m$, where M is a very large integer; if $r < \overline{k}$ $m = 0$. The fitness function can be expressed as $f = 1/(R + M \times m)$. For convenience, capacity of the vehicles is sameness, and the maximum distance that each vehicle can travel are equal, they can be denoted respectively as Q and L

.

3.2. Improved differential evolution operators

Differential Evolution grew out of Ken Price's attempts to solve the Chebychev Polynomial fitting Problem that had been posed to him by Rainer Storn in 1996. The basic steps are as fellows:

(1) Initiation population: We adopted an integer coding as section 3.1, the initial population is generated by random generator and the number of individual is NP, each individual is an N-dimensional solution vectors.

(2) Mutation operation: The chromosome of offspring generate by parent gene difference, mutation is an operation that adds a vector differential to a population vector of individuals, according to the following equation:

$$v = Chrom(c,:) + F * [Chrom(a,:) - Chrom(b,:)] \quad , (10);$$

where a, b, c are generated randomly, the scaling factor F is a constant from [0,2].Because we adopted an integer code, and each chromosome represents a sequence of the customers, each gene stands for a customer, when the offspring gene oversteps the range, we must consider an auxiliary operator based on integer order criterion. For the largest gene of offspring gives the largest customer ordinal number N, the second evaluated as $N-1$, the rest may be deduced by analogy, we can prove that this operator equates to an affine transform, for example, the offspring chromosome is [-7,1,-5,2,-3, 0, 3, 5], the number of the customers is 8, we obtain the offspring chromosome that is [1, 5, 2, 6, 3, 4, 7, 8].

(3)Crossover operation: Following the mutation operation, crossover operation is applied to the population. Crossover operation is employed to generate a trial vector by replacing certain parameters of the target vector by the corresponding parameters of a randomly generated donor vector. That is

$$trial(j)^{G+1} = \begin{cases} v(j)^{G+1}, & rand(j) \le CR \ \ or \ \ j = randn(i) \\ Chrom(i,j)^{G}, rand(j) > CR \ \ and \ \ j \ne randn(i) \end{cases} , \quad (11)$$

where G is the number of current iteration, $CR \in [0,1]$ is the crossover probability factor. In order to improve the population's diversity and the ability of breaking away from the local optimum, we present a new self-adapting differential evolution algorithm, the key factor is the crossover probability (CR) is time varying, it changes from small to large with iteration number. i.e.

$$CR = CR_{\min} + G * \frac{CR_{\max} - CR_{\min}}{MAXGEN}, \qquad (12)$$

where $CR_{\min}$ is the proposed minimum crossover probability, and $CR_{\max}$ is the maximum crossover probability, G is the number of current iteration, $MAXGEN$ is the number of maximum iteration. In the early stage of evolution, the crossover probability is smaller, which can improve the global searching capability; in the later stage of evolution, the crossover probability is larger, which can improve the local searching capability.

(4) Estimation and Selection operation: The parent is replaced by its offspring if the fitness of the offspring is better than that of its parent. Contrarily, the parent is retained in the next generation if the fitness of the offspring is worst than that of its parent, according to the following equation:

$$Chrom(i,:)^{G+1} = \begin{cases} trial^{G+1}, & f(Chrom(i,:)^{G+1}) < f(trial^{G}) \\ Chrom(i,:)^{G}, otherwise \end{cases} \quad ,(13)$$

Usually, the performance of a DE algorithm depends on three variables: the population size NP, the scaling factor F and the crossover probability CR.

4. COMPUTATIONAL EXPERIMENTS

The improved differential evolution algorithm described in the previous section is coded in Matlab language and applied to the 8-customers vehicle routing problem with simultaneous delivery and pick-up. There are three vehicles in the depot, capacity of each vehicle is 8 tons, delivery and pick-up demands of the 8 customers are listed in Table 1, the distances between customers and depot is listed in Table 2. Parameters for the IDE are as follows: NP is 40, $MAXGEN$ is 200, F is 0.5, $CR_{\min}$ =0.3, $CR_{\max}$ =0.9; and the maximum distance L is 400 kilometers.10 independent trials are carried out to evaluate the average performance. There are 10 independent trials are carried out to evaluate the average performance. The results of simulations are presented in Table 3, and the best solution obtained is showed in Figure 4.

Table 1 The delivery and pick-up demands of customers.

Customer i	1	2	3	4
Delivery demands	2	1.5	4.5	3
Pick-up demands	3	1	2	2

Customer i	5	6	7	8
Delivery demands	1.5	4	2.5	3
Pick-up demands	3	4	1.5	3

Table 2 The distance matrix

i / j	0	1	2	3	4
0	0	40	60	75	90
1	40	0	65	40	100
2	60	65	0	75	100
3	75	40	75	100	100
4	90	100	100	100	0
5	200	50	100	50	100
6	100	75	75	90	75
7	160	110	75	90	75
8	80	100	75	150	100

i / j	5	6	7	8
0	200	100	160	80
1	50	75	110	100
2	100	75	75	75
3	50	90	90	150
4	100	75	75	100
5	0	70	90	75
6	70	0	70	100
7	90	70	0	100
8	75	100	100	0

Table 3 The average computational results of iterating 200.

Vehicles	Distance (Km)	Percent-age of reaching optimal solution	Comput-ational time (second)	Routes
3	790	100%	1.5160	0-3-5-1-0; 0-6-7-2-0; 0-8-4-0;

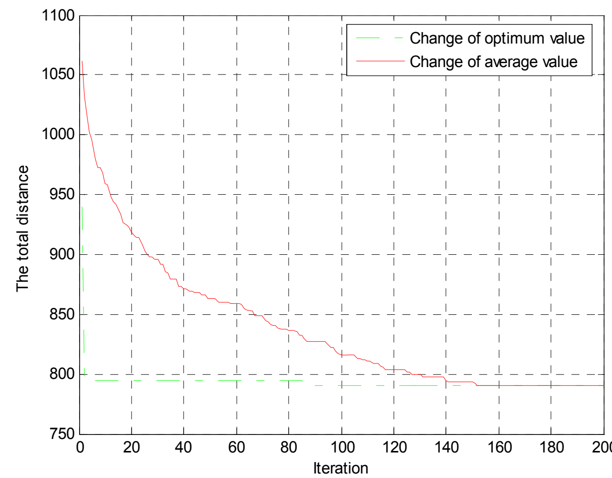

Figure 4 The results of iterating 200.

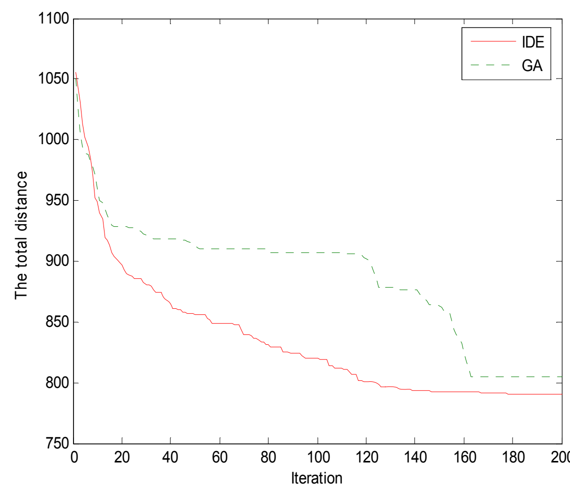

Figure 5 The comparison of IDE with GA.

The program was coded in Matlab language and simulations are performed on a personal computer with 3.06MHz Pentium 4 processor and 1G of RAM, the runtime do not exceed two seconds. A run of the improved differential evolution algorithm shows that the best

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE
IEEE COMPUTER SOCIETY

operational plan is

Vehicle 1: depot—3—5—1—depot, the delivery and pick-up demand both are 8 tons and achieve full loads, the route distance is 215kms.

Vehicle 2 : depot—6—7—2—depot, the delivery demand is 8 tons, and achieve full loads, the pick-up demand is 6.5 tons, the route distance is 305kms.

Vehicle 3: depot — 8 — 4 — depot, corresponding delivery demand is 6 tons, full loads ratio is 75%, pick-up demand is 5 tons, the route distance is 270kms.

The total distance traveled by the three vehicles is 790kms. From the computational results of Figure 4, the results obtained by our IDE are robust, the distance approximate balance of the three routes, we explain this from the viewpoint of mutation operation and crossover operation we carried out.

To verify the performance of the proposed improved differential evolution algorithm, the above example system was repeatedly by two hundred times by the IDE and GA [9] respectively. The parameters for GA are included as follows: NP is 40, $MAXGEN$ is 200, probability of crossover operation is 0.95, probability of mutation is 0.01, and the maximum distance L is 400 kilometers. The total distance traveled by three vehicles is 805kms, and the operational plan is

Vehicle1: depot—4—5—8—depot. Vehicle 2: depot —2—7—6—depot. Vehicle 3: depot—1—3—depot. And the runtime is 1.5320 seconds, which is longer than that of our proposed method. The comparison results between two algorithms were shown in Figure 5, obviously, the total distance traveled of IDE reduced 15kms, and the convergence speed is more rapid and the robustness is better than GA.

5. CONCLUSIONS

This paper contributed to vehicle routing problem with simultaneous delivery and pick-up in the following respects: (a) an integer programming mathematic model of VRP-SDP was proposed for finding the optimal solutions, which described the relationship between VRP-SDP and other vehicle routing problems ; (b) an improved differential evolution algorithm to solve the vehicle routing problem with simultaneous delivery and pick-up was presented, focusing on total traveled distance minimization, in the operation process, decimal permutation encoding was used to represent solution and penalty function was designed to eliminate illegal solutions, more importantly, the mutation operator、crossover operator and selection operator were used as improved differential evolution operators to prevent premature convergence and accelerate searching procedure; (c) the effectiveness of the improved differential evolution algorithm was shown by some numerical examples.

ACKNOWLEDGEMENT

This research was supported by the Specialized Research Fund for Doctoral Program of Higher Education of China under Grant 20050532029.

REFERENCES

[1]Stock J R. *Reverse Logistics*. Council of Logistics Management, Oak Brook , IL,1992

[2] Min H. “The multiple vehicle routing problem with simultaneous delivery and pickup points”. *Transportation Research* A. 23, pp.377–386. 1989.

[3]Dethloff, J. *“Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up”*. OR Spektrum.23, pp.79–96.

[4]Angelelli, E. Mansini, R. “A branch-and-price algorithm for a simultaneous pick-up and delivery problem”, *Working Paper, Article presented at the EURO/INFORMS Meeting,* 2003.

[5]Tang FA, Galvao RD. “Vehicle routing problems with simultaneous pick-up and delivery service”, *Journal of the Operational Research Society of India (OPSEARCH)* 39, pp.19－33. 2002.

[6]Tang FA, Galvao RD. “A tabu search algorithm for the vehicle routing problems with simultaneous pick-up and delivery service”, *Computer & Operation Research.* 33, pp.595-619. 2006.

[7]Baker Barrie M, Ayechew M. A. “A Genetic Algorithm for the Vehicle Routing Problem”, *Computers & Operations Research.* 30, pp.787-800,2003.

[8]Christian Prins. “A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem”, *Computers & Operations Research.*31, pp.1985-2002, 2004.

[9]Gen, M.Cheng, R. *Genetic Algorithms &Engineering Optimization*. New York: Wiley, 2000.

[11]Storn R. ‘Differential evolution design of an IIR-filter’, *Proceedings IEEE Conference Evolutionary Computation,Nagoya, Japan: IEEE,* pp.268-273, 1996.

[12]DEHomepage. http://www.icsi.berkeley.edu/~storn/ code.html