An Improved Genetic Algorithm for Vehicle Routing Problem

Jiashan Zhang(✉)

Chongqing Vocational Institute of Engineering, Chongqing 402260, China
zh_jiashan@163.com

Abstract. Evolutionary algorithms, including genetic algorithm, usually appear premature convergence. In this paper, new crossover and mutation operators are introduced. This paper addresses an application of improved genetic algorithms (IGA) for solving the Vehicle Routing Problem (VRP). After the introduction of new genetic operators, diversity of population becomes abundant. The ability of global search is obviously improved in new algorithm. Simulations indicate that new genetic algorithm is competitive with other modern heuristics.

Keywords: Premature convergence · Genetic operator · Global search · Simulations

1 Introduction

In the optimization of distribution networks, the vehicle routing problem (VRP) [1, 2] remains a challenging issue. Traditional exact algorithms [3] play a very important role as well as its limitation. They can only solve VRPs less than 50 nodes, not solve large-scale VRPs. Approximate algorithms, which aim at finding approximate solutions in the finite time, are applied to large-scale instances widely. For example, [4] solves VRP based on tabu search. [5] applies simulated annealing to VRP. An improved particle swarm optimization (PSO) is proposed to solve the vehicle routing problem in[6]. Jun Zhang [7] proposed an Ant Colony Optimization for VRP with Time Windows. An adaptive memory strategy for VRP is proposed in [8]. Since genetic algorithm (GA) [9, 10] was put forward, it has gained extensive attention of researchers. It is robust and flexible. It has been applied to many combinatorial problems. For example, genetic algorithm is used to optimize the capacitated clustering problem [11]. However, the general Genetic Algorithm usually appear premature convergence. Numerous successful applications intensively favor improved algorithm. In this paper, a competitive improved genetic algorithm (IGA), in which new crossover and mutation operators are designed, is proposed.

2 Formulation for VRP

There are m vehicles in the distribution center, which are used to delivery cargo for n customers $(v_1, v_2, \ldots, v_n)$. Demand level of each customer q_i is known. All vehicles are required to start from distribution center. All vehicles are required to return back to

J. MacIntyre et al. (Eds.): SPIoT 2020, AISC 1282, pp. 163–169, 2021.
https://doi.org/10.1007/978-3-030-62743-0_23

distribution center after finishing distribution. Every customer's demands are satisfied by just one vehicle' service. Here, let the total distance traveled represent distribution cost. To find an path with the minimum cost is this paper objective.

Set 0–1variables as follows

$$x_{ijk} = \begin{cases} 1, & \textit{if} \text{ vehicle k visits customer j after i} \\ 0, & \textit{else} \end{cases} \qquad y_{jk} = \begin{cases} 1, & \textit{if} \text{ vehicle k visits client j} \\ 0, & \textit{else} \end{cases}$$

The objective function is:

$$\min z = \sum_i \sum_j \sum_k c_{ij} x_{ijk} \tag{1}$$

S.t.

$$\sum_{i=1}^{n} q_i y_{ik} \le Q, \qquad k = 1, 2, \ldots, m \tag{2}$$

$$\sum_i \sum_j d_{ij} \cdot x_{ijk} \le \mathrm{L}, \quad k = 1, 2, \ldots, m \tag{3}$$

$$\sum_{k=1}^{m} y_{ik} = \begin{cases} 1, & i = 1, 2, \ldots, n \\ m, & i = 0 \end{cases} \tag{4}$$

$$\sum_{i=0}^{n} x_{ijk} = y_{jk}, \qquad j = 1, 2, \ldots, n; \quad k = 1, 2, \ldots, m \tag{5}$$

$$\sum_{j=0}^{n} x_{ijk} = y_{ik}, \qquad i = 1, 2, \ldots, n; \quad k = 1, 2, \ldots, m \tag{6}$$

3 The Proposed IGA for VRP

3.1 Framework

In this paper, calculate an initial population at first, i.e. the first generation. The paper suppose that the initial population is made up of n individuals. Population size (popsize) should be appropriate. For each individuals, there is a fitness value. We can compute them, and rank individuals according to fitness value. Then, we choose a pair of individuals with high fitness value(parents), and apply cross operation on them. Two new individuals are get, called children. Subsequently, we apply mutation operation to the children produced. The algorithm stops until termination condition is satisfied.

The improved algorithm framework is shown below:

```
Initialization
Randomly generate the initial chromosome population
Repeat
k=1
  For p =l:mdo
Choose two chromosomes Si,Sjfrom previous generation
Produce the chromosome  Sn based on the proposed crossover；
   If fit(Sn)> fit(Si), and fit(Sn)>fit(Sj)
            Add Sn to population
   Else
            Abandon Sn
        Excute mutation operation;
End for
   k=k+ 1
  Until (termination conditions)
```

3.2 Coding and Fitness Function

Coding plays an important role. For each VRP solution, good coding is helpful to identify the number of vehicles [12]. Actually, a chromosome I(n) simply is corresponding to a sequence S of nodes. For example, there are 9 customers, i.e. 9 nodes. Randomly generated a chromosome: 5-3-8-2-7-1-6-9-4. We can divide the chromosome 5-3-8-2-7-1-6-9-4 into three parts: 0-5-3-8-2-0, 0-7-1-6-0, and 0-9-4-0. Each part stands for an feasible route. For each vehicle, the total demand of customer nodes must less than the capacity of vehicle. Total number of vehicles required does not exceedm. Then the solution is legal; otherwise, it is illegal.

To measure quality of solution, each chromosome has a fitness value assigned. Here, This paper choose the total distance of travelling for all vehicles as fitness value.

$$fit(Si) = \frac{1}{total\_distance(Si)}$$

3.3 Crossover

Crossover is performed with two chromosomes, called parent chromosomes. The crossover aims to generate off springs, child chromosomes. It is a probabilistic process. In most cases, it is possible to get one child or two children from two parents. It leads to take information from both parents and transfer it to the children, which accelerates search process. To prevent premature convergence, the crossover probability p_c is not constant. When the iteration is lower than ten percent of the prespecified number of generation N_{cmax}, the crossover probability remains fixed, or shown below.

$$p_c = \begin{cases} k_1, & f_c \le f_{avg} \\ \frac{k_1(f_{\max} - f_c)}{f_{\max} - f_{avg}}, & f_c > f_{avg} \end{cases} \tag{7}$$

Where k_1 is a constant, and $0 < k_1 < 1$.

Let the two parent solutions be $P_1 = (123456789)$, $P_2 = (452187693)$. The crossover procedures are shown below.

(1) Choose two positions randomly in the parents, see Fig. 1(a).
(2) Scramble the order of genes randomly. Repeat the operation, see Fig. 1(b).
(3) Exchange two substrings, see Fig. 1(c) and (d).

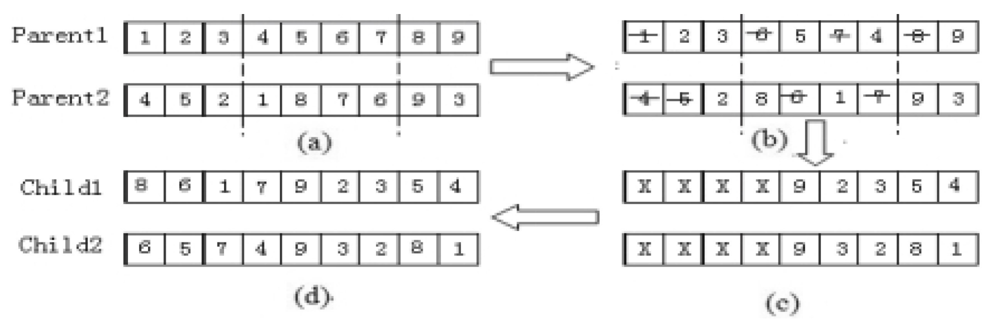

Fig. 1. Crossover operators

After crossover, calculate fitness values of Parent1, Parent2, Child1 and Child2 and evaluated them. Choose two solutions with better fitness value, labeled C1, C2. They are preserved into next iteration.

3.4 Mutation

Mutation happens with a small probability. In this paper, when the iteration is lower than ten percent of the prespecified number of generation N_{cmax}, the mutation probability remains fixed, or shown below.

$$p_m = \begin{cases} k_2, & f_m \le f_{avg} \\ \frac{k_2(f_{\max} - f_m)}{f_{\max} - f_{avg}}, & f_m > f_{avg} \end{cases} \tag{8}$$

The mutation operator *Inversion* [12] is a widely used. See procedures as follows.

1. Randomly choose mutation points: $P_1 = 2$, $P_2 = 6$.
2. Reverses the segment between these two mutation points.
3. Repeated reverse operation for n/10 times.

Inversion operation is shown below (Fig. 2).

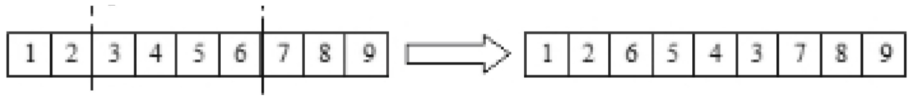

Fig. 2. Inversionmutationoperators

4 Experiments

In this section, the improved genetic algorithm (IGA) proposed has been executed on an Intel Pentium 5 computer, which is equipped with 4 GB memory, running 10 times. Our simulation is based on the instance China Traveling Salesman Problem (CTSP).

Parameters are set as below:

Popsize = 50, $N_{cmax} = 1000$, probability of crossover operation $p_c = 0.90$, and mutation probability: $p_m = 0.005$. The results of computation are shown below (Table 1).

Table 1. Comparison of general genetic algorithm and improved genetic algorithm

Number	General genetic algorithm	Improved genetic algorithm
1	15460	15415
2	15745	15480
3	15871	15736
4	15457	15381
5	15743	15590
6	15621	15471
7	15634	15482
8	15449	15381
9	15478	15454
10	15452	15381

The best solution obtained is showed in Fig. 3.

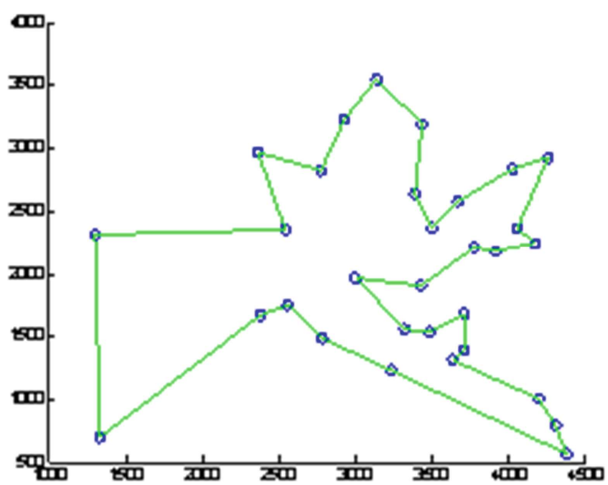

Fig. 3. Best solution found for problem

The also compared the general GA, IGA and other methodes. The results of simulation are presented in Table 2.

Table 2. Comparison of different algorithms

Method	Best solution	Number of iterations
General ACA	15512	234
Or-opt method [10]	15437	–
General GA	15449	192
Improved genetic algorithm	15381	627

5 Conclusion

To solve Vehicle Routing Problem, an improved genetic algorithm (IGA) is proposed in this article. An effective crossover operator is introduced, which helps to improve the global searching ability. Compared with general GA, the crossover probability varies adaptively, which prevents premature convergence of GA. Inversion mutation gradually add some new characteristics to the population. The improved genetic algorithm can effectively solve a variety of vehicle routing problems and retain optimum.

Acknowledgements. Foundation item: Supported by Chongqing Education Department Science and Technology Project (No. KJ1603201), and Chongqing Education Department project (No. 173245).

References

1. Christofides, N.: The vehicle routing problem. Oper. Res. **59**(2), 55–70 (2010)
2. Paolo, T., Daniele, V.: Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Appl. Math. **123**, 487–512 (2002)
3. Laporte, G.: Thevehicle routing problem: an overview of exact and approximate algorithms. Eur. J. Oper. Res. **59**, 345–348 (1992)
4. Gloyer, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. **13**, 533–549 (1986)
5. Metropolis, N., Rosenbluth, A., Rosenbluth, M., et al.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1092 (1953)
6. Ai, T.J., Kachitvichyanukul, V.: A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. Comput. Oper. Res. **36**, 1693–1702 (2009)
7. Gajpal, Y., Abad, P.: An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. Comput. Oper. Res. **36**, 3215–3223 (2009)
8. Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T.: An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups anddeliveries. Eur. J. Oper. Res. **202**, 401–411 (2010)
9. Holland, J.H.: Adaptation in Natural and Artificial System. University of Michigan Press, Ann Arbor (1975)

10. Jong De, K.A.: An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph.D. Dissertation, University of Michigan, U.S.A. (1975)
11. Shieh, H.M., May, M.D.: Solving the capacitated clustering problem with genetic algorithms. J. Chin. Inst. Ind. Eng. **18**, 1–12 (2001)
12. Liu, G.: Improved ant colony algorithm for solving VRP problem under capacity and distance constrains. J. Guangxi Univ. Nationalities (Nat. Sci. Edn). **02**, 51–53 (2010)