

Obtaining the Most Beautiful Route of Map Navigation Based on Genetic Algorithms

Zhang Kai

School of Science and Engineering
East China Normal University
Shanghai, China
51131211017@ecnu.cn

Liu Yao

School of Science and Engineering
East China Normal University
Shanghai, China
liuyao@cc.ecnu.edu.cn

Wu Ping *

School of Science and Engineering
East China Normal University
Shanghai, China
kaikai_z@126.com

* Corresponding Author

Abstract—This paper particularly describes a Multi-Objective Chromosome-length-variant Genetic Algorithm (MCGA) as an algorithm for finding the most beautiful route of map navigation, compared to the conventional navigation which can only obtain the shortest route. The critical task to this problem is how to deal with multi-objective optimization and the length-variant chromosome. We proposed a dual stage linear polymeric fitness function which can well balance the multi-object and optimize the convergence effect. Cross and mutate operators are modified to a new set of insert, delete and mutate operators which can potentially process all possible route in navigation. Additionally, a sort operator is added to the algorithm to enhance convergence rate. Illustrative experiment results of how the MCGA obtains the most beautiful route from Shanghai to Chengdu are presented. The results also show that users have the ability to choose any candidate route primarily depending on their favor.

Keywords—the most beautiful route, GA, multi-object, length-variant chromosome

I. INTRODUCTION

Map Navigation has become increasingly popular since users can obtain the shortest or most convenient routes using general navigations in a short time period only by entering a starting point and a destination [1]. Yahoo Laboratory raised their research of shortest paths to happiness in cities which make the journey more pleasant and interesting [2]. In this way, the journey is not only focus on the shortest route to the destination, but also focus on whether the route will pass beautiful sceneries. The research is still in the experimental stage and has not been applied to practice. Due to the huge amount of relevant data and the complexity of multi-objective optimization, it's difficult to develop an automatic high-performance algorithm to obtain a beautiful route with map navigation.

This paper describes a novel method based on Genetic Algorithm (GA) to obtain a beautiful route between two

places. For this problem, two issues can be solved by using genetic algorithm: Firstly, while both of length and scenery of the route should be optimum, it should be treated as a Multi-Objective Genetic Algorithm (MOGA). MOGA optimization approached into a weighted single objective problem in [3]. Weights are assigned according to the importance of each object. The solution of a global optimized objective function formed by positive weighted vector is definitely a Pareto optimal solution [4]. The authors introduced the Pareto efficiency criterion based on genetic algorithm for signatures in [5]. A multi-objective optimization approaches to camera and light source placement in [6]. In [7], the authors introduced a binary tree method to weight order and majority of multi-object. Secondly, with the uncertainty of the result routes, the result sequences are most likely to be variable-length. In [8], the authors controlled the length of chromosomes with special crossover and mutation operators. Hidden and observation nodes method were designed to deal with variable length chromosome in [9]. Moreover, a novel model for navigation experience sharing presented in [10].

With previous approaches to genetic algorithm and map navigation, we proposed a Multi-Objective Chromosome-length-variant Genetic Algorithm (MCGA) which can process multi-objective optimization and variable length chromosome problem by designing linear polymeric functions and special evolutionary operators.

II. BEAUTIFUL ROUTE PROBLEM

An undirected map $M = (V, R, E)$ comprises a set of nodes $V = \{v_i\}$ (for $i=1,2,\dots,N$), a set of ranks $R = \{r_i\}$ (for $i=1,2,\dots,N$) representing each node's scenery level and a set of edges $E \in V \times V$ connecting nodes in V . The cost between two different nodes (from node v_i to node v_j) is a non-negative number p_{ij} . The rank of each node is also a non-negative number r_i . The route from node v_i to v_j can be represented as a sequence of node $(v_i, v_k, v_l, \dots, v_m, v_j)$ which no node appears more than twice. This route's rank

$R = \sum_{v_i}^{v_j} v_n$ is the sum rank of nodes which the route consists. For the example map given in Fig .1, numbers in parentheses are corresponding nodes' ranks. (1,2,4,9,10) is a possible route from node 1 to node 10. The problem is to find the most beautiful route between two given nodes which have the minimum sum length and maximum sum rank.

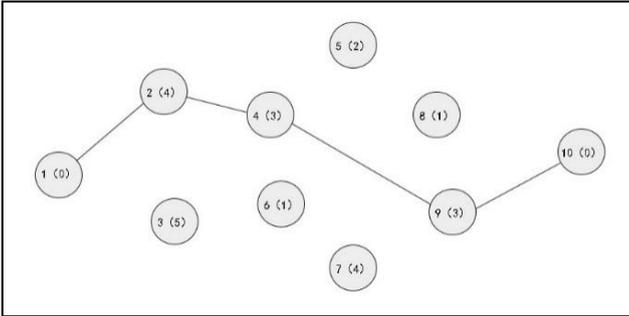


Figure 1. A map with 10 nodes and a possible route

III. LENGTH-VARIANT CHROMOSOME ENCODING AND OPERATION

Unlike the traditional solutions based on genetic algorithm (such as the traveling salesman problem) which encodes the path chromosome with fixed-length characters [11], number of result nodes to this problem is variable. Additionally, a random sequence of nodes rarely corresponds to a specific route [12]. All of these make critical impact to encode route to chromosome. To overcome such difficulties, we encoded the route with nodes' serial numbers and design additional operators to maintain the variable length chromosomes. Other route optimization issues (Partheno-Genetic Algorithms, messy Genetic Algorithms) have also been demonstrated the superiority of using serial-number-coded compared to binary-coded [13].

In order to make the length of chromosomes variable during evolution and also serve a similar capacity of crossover and mutate operator, we presented the novel insert, delete and mutate operators. These operators act on the individuals of parent generation.

Insert Operation: In a generation of genetic algorithm. Assuming we get a chromosome route set R, we let the

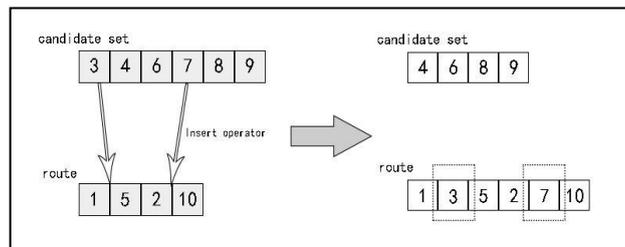


Figure 2. The insert operator

nodes which are not in R be the candidate set. If the insert probability value meets the requirement, we choose a random number of nodes from the candidate set, and insert

them at set R's random location. The location should not be in front of the first node nor behind the last node. For the example given in Fig .2, we define a ten-node map. A possible parent chromosome route is (1,5,2,10). After the insert operation, a possible child chromosome route is (1,3,5,2,7,10).

Delete Operation: Similar to the insert operation, if the delete probability value meets the requirement, delete a random number of nodes. The deleted nodes should not include the first node nor the last node which represent the starting point and the terminal point. For the example given in left of Fig .3, after the delete operation, a possible child chromosome route is (1,3,2,10) compared with the parent chromosome route (1,3,5,2,7,10).

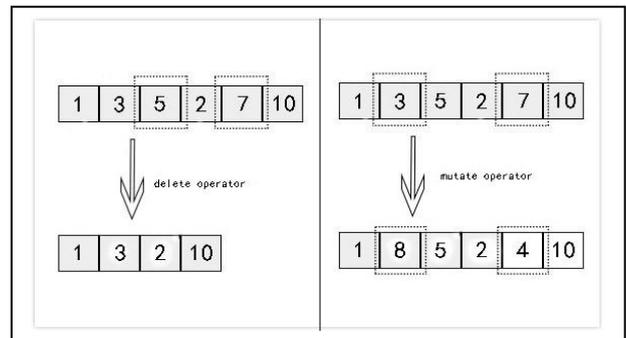


Figure 3. The delete operator

Mutate Operation: Unlike the traditional mutate operation, we modified it with a new method. For the current chromosome route, we build a candidate set as it was built in insert operation. If the mutate probability value meets the requirement, choose a random number of nodes in candidate set to replace the nodes in the current chromosome route. Also the replaced nodes should not include the first node nor the last node which we mentioned in the delete operation. For the example given in right of Fig. 3, after the mutate operation, a possible child chromosome route is (1,8,5,2,4,10) compared with the parent chromosome route (1,3,5,2,7,10).

In the course of the genetic algorithm, it is obvious that the above operators are able to reach any solution in the entire solution space. And in the search process, a child chromosome generated only depends on the parent chromosome which conforms to Markov Criterion.

Sort Operation: In order to make the algorithm converge to a good result as soon as possible and reduce unnecessary computing to candidate set, we proposed a sort operation. It operators on the chromosome routes after insert, delete and mutate operations. It sorts the nodes in the route according to the distances to the terminal node. It can be simply proved that the route which is sorted is always better than which is unsorted because the length of sorted route is shorter but the rank remains unchanged. The operator helps produce better solutions and reduce the search space effectively. For the example given in Fig .4, after the sort operation, the result route is (1,4,3,6,10) compared with the unsorted route (1,6,4,3,10).

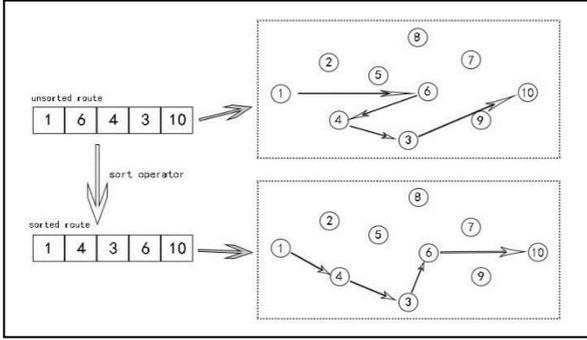


Figure 4. The sort operator

Probability: In the traditional genetic algorithm, crossover and mutation probabilities remain unchanged in the whole operation process. But many previous research has shown that they have a large impact on the performance of genetic algorithm[14][15]. We set the probability value of operations as a self-adaption value according to other scientists' research[16]. They will change in different stages of the algorithm to optimize the operators' effect.

IV. MULTI-OBJECTIVE FITNESS FUNCTION

The route chosen to this problem should meet two objectives: the shortest length and the highest rank. Because of pivotal role that the fitness function plays in GA[17], we designed a novel two-state fitness function. The initial state of the fitness function is designed with the average weight method. We assume a chromosome route is $\{v_0, v_1, \dots, v_i, \dots, v_n\}$.

Defines the following variable:

P_i : length from node No. i to node No. $(i+1)$.

$\sum_{i=0}^n P_i$: sum length of a chromosome route.

P_{\min} : length for start node v_0 to end node v_n .

$rank_i$: rank of each node.

$\sum_{i=0}^n rank_i$: sum rank of a chromosome route

The initial state of fitness function is given as follows:

$$fit_{ini} = \frac{1}{2\pi} \left(\frac{\pi}{2} + \arctan \left(\left(\sum_{i=0}^n P_i - c_1 \times P_{\min} \right) \times c_2 \right) \right) + \frac{1}{2} \left(1 - \sin \left(\frac{\pi}{2} \times \frac{\sum_{i=0}^n rank_i}{5 \times n} \right) \right) \quad (1)$$

The parameter C_1 represents the farthest distance users can withstand. The parameter C_2 maintains the data evenly scattered in the range of trigonometric functions. From the (1), length and rank of route each share half of the proportion. The range of the initial state of fitness function's value is (0, 1), and it equals 1 ideally. In the equation, both the value of P and R equal 0.5. The values of the fitness function are single-valued and continuous. It conforms to the general criteria of fitness function structure.

To avoid the two deceptive problems of genetic algorithm [18], we proposed a function to revise the former fitness function. We defined it as a final state of fitness function. It is given as follows.

$$fit_{fin} = \begin{cases} 1 - 0.5 \times \left(\frac{fit_{ini} - fit_{\min}}{fit_{avg} - fit_{\min}} \right)^{\frac{1}{4}}, & if (fit_{ini} \leq f_{avg}) \\ \frac{1}{1 + \left(\frac{fit_{ini} - fit_{\min}}{fit_{avg} - fit_{\min}} \right)^4}, & else \end{cases} \quad (2)$$

The parameter fit_{ini} is the value of equation in (1).

fit_{avg} and fit_{\min} are the average and minimum values of the current generation of chromosomes' fitness function. We can acquire the graphic of the function. It shows in Fig.5.

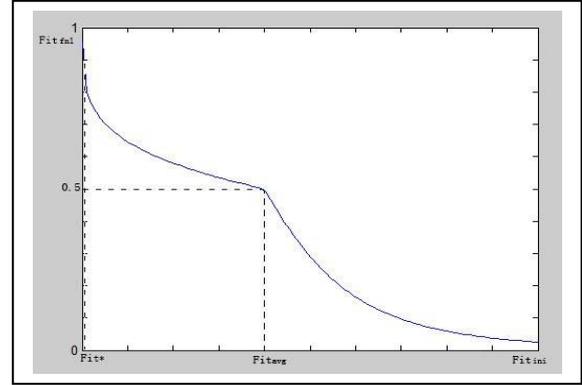


Figure 5. The graph of fitness function

As the Fig .5 shows, the range of the final state of fitness function's value is (0,1), and the value is not limited to a small subdomains of the range. The fitness value changes much more sensitively if individuals are more closes to optimal solution Fit^* . The individuals below the average value go down faster. The significance of this design is the algorithm can eliminate individuals which have lower fitness values as soon as possible in the early stage and it can also widen the gap between the individuals close to the optimal solution. The function helps algorithm to make sensitive choice as an encourage means. It also helps avoid a suboptimal solution effectively.

V. PROCEDURE TO OBTAIN BEAUTIFUL ROUTE

The pseudo-code as follows:

Input: map data and relevant parameter

Output: optimal solution which possibly is the most beautiful route

```

Main ():
  Initial Data and declare functions;
  Set elitist = (v0, vn);
  While i < max number of generation:
    Calculate the fitness of each chromosome;
    While j < max size of population:
      Select operator ();
      Insert operator ();
      Delete operator ();
      Mutate operator ();
      Sort operator ()
    j ++;
  Save the best chromosome as a result;
  If the result is better than the elitist:
    Set elitist = result;
  i ++;
  Print elitist;
End

```

VI. NUMERICAL EXAMPLES

To verify and analysis the performance of Multi-Objective Chromosome-length-variant Genetic Algorithm, we wrote a Python language program for the algorithm. We set Shanghai as the starting point and Chengdu as the terminal point in the simulation and choose 50 cities along the route from Shanghai to Chengdu as sceneries. We collected the distance data between each city from Yahoo map navigation. The rank of each city is defined by information from Google. The rank has 5 levels from 1 to 5 with 1 stands for the least beautiful city and 5 stands for the most beautiful city. We defined both the rank of the starting city and the end city as 0. The program was ran on the windows system. After observing a large number of experimental results, we assigned the parameters as shown in Table I.

TABLE I. PARAMETERS OF ALGORITHM

Parameters	Value
Size of population	50000
Max number of evolution	100000
Probability value of operations	Self-adaption

The results obtained using the MCGA are shown in Table II. The distance from node 0 (Shanghai) to node 49 (Chengdu) is 1963.2 km. Both the rank of node 0 and node 49 are 0. According to the experimental results, the rank will increase with the increase of corresponding route's distance. Users can choose the route of either high rank with long route or low rank with short route.

TABLE II. TEN RESULTS OF ALGORITHM

Result Routes	Distance(km)	Rank
0, 12, 49	1971.9	2
0, 6, 49	1973.8	4
0,6,7,30,32,49	1994.9	11
0,1,2,41,17,49	2267.7	16
0,2,15,17,33,37,49	2285.3	24
0,5,2,15,28,33,49	2344.2	24
0,3,5,2,15,17,33,37,49	2453.0	33
0,5,2,15,26,33,49	2460.3	24
0,5,9,17,28,37,49	2484.2	25
0,5,6,7,40,19,45,17,33,35,36,49	2632.0	30

Fig .6 shows the route (0,3,5,2,15,17,33,37,49) in the real map. The cities on the route are all famous tourist cities in China.

To make experimental results more convincing, we compared the results with random generated results and artificial selection results. The average results of 10 experiments are given in Table III. It is obviously that result of MCGA enjoys a distinct advantage.

From the results above, our goal which is obtain the most beautiful route has been accomplished. The result route shown in Fig .6 is much more beautiful than the route just from Shanghai to Chengdu. Despite being about 20% longer in length, users preferred taking routes which have more beautiful sceneries. They also can take more sceneries route with longer length. Because of the tremendous amount of relevant data, it is difficult for users



Figure 6. A route in map

to obtain the most beautiful route manually which the algorithm can do automatically. From the set of result routes we got, users can choose the specific route (including the very city they like) depending on their preferences.

TABLE III. RESULTS OF THREE WAYS

Method	Average distance	Average rank
MCGA	2286.73	19.3
Random generated	10357.41	48.7
Artificial selection	2853.72	17.5

VII. CONCLUSIONS AND FUTURE WORK

This paper presents Multi-Objective Chromosome-length-variant Genetic Algorithm to obtain the most beautiful routes in map navigation. This problem is researched hardly ever. We proposed new operators to deal with variable length chromosome and a novel fitness function to weigh multi-object. From the experimental results, we know that the algorithm can achieve the original design requirement and also obtain the appropriate and effective solutions. The most beautiful route navigation makes the trip more interesting than the traditional map navigation. The solution of this problem will not be one result but a set, and users can choose one from it depending on their preferences. Additionally, more requirements can be added into consideration (such as emotion, weather) in future researches. With more data being used in the algorithm, there is room for improvement in the calculating time.

ACKNOWLEDGMENT

This research is partly supported by National High Technology Research and Development Program of China No.2013AA01A211.

REFERENCES

- [1] Takata, Y., and Watanabe, H. "Development of the Map Navigation System Targeting Users," in: 2013 International Conference on Cyberworlds (CW), pp. 385-385. IEEE. (2013)
- [2] Quercia, D., Schifanella, R., and Aiello, L. M. "The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city," in: Proceedings of the 25th ACM conference on Hypertext and social media, pp. 116-125. ACM. (2014)

- [3] Gao, Y., Shi, L., and Yao, P. "Study on multi-objective genetic algorithm," in: Proceedings of the 3rd World Congress on Intelligent Control and Automation, Vol. 1, pp. 646-650. IEEE. (2000)
- [4] Fonseca, C. M., and Fleming, P. J. "Genetic Algorithms for Multi objective Optimization: Formulation Discussion and Generalization," in: ICGA, Vol. 93, pp. 416-423. (1993)
- [5] Calonder, M., Lepetit, V., and Fua, P. "Pareto-optimal dictionaries for signatures," in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3011-3018. IEEE. (2010)
- [6] Garg, R., Indu, S., and Chaudhury, S. "Camera and Light Source Placement: A Multi-Objective Approach," in: Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 187-191. IEEE. (2011)
- [7] Nguyen, A., Urli, T., and Wagner, M. "Single-and multi-objective genetic programming: new bounds for weighted order and majority," in: Proceedings of the twelfth workshop on Foundations of genetic algorithms XII, pp. 161-172. ACM. (2013)
- [8] Hong, Y. Y., and Chen, P. H. "Genetic-based underfrequency load shedding in a stand-alone power system considering fuzzy loads," in: IEEE Transactions on Power Delivery, 27(1), pp. 87-95. (2012)
- [9] Moradi Kordmahalleh, M., Gorji Sefidmazgi, M., Homaiyar, A., KC, D. B., and Guiseppi-Elie, A. "Time-series forecasting with evolvable partially connected artificial neural network," in: Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion, pp. 79-80. ACM. (2014)
- [10] Karimi, H.A., Benner, J.G. and Anwar, M. "A Model for Navigation Experience Sharing through Social Navigation Networks," in: 2011 IEEE International Conference on Information Reuse and Integration (IRI), pp. 557-560. IEEE. (2011)
- [11] Miller, J., Kim, S. I., and Menard, T. "Intelligent Transportation Systems Traveling Salesman Problem (ITS-TSP)-a specialized tsp with dynamic edge weights and intermediate cities," in: 2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 992-997. IEEE. (2010)
- [12] Bellman, R. "On a routing problem," RAND CORP SANTA MONICA CA. (1956)
- [13] Goldberg, D. E. "Real-coded genetic algorithms, virtual alphabets, and blocking," Urbana, 51, 61801. (1990)
- [14] Osaba, E., Diaz, F., Onieva, E., Carballedo, R., Perallos, A., Osaba, E., and Lee, V. C. S. "AMCPA: A Population Metaheuristic With Adaptive Crossover Probability and Multi-Crossover Mechanism for Solving Combinatorial Optimization Problems," International Journal, 14, Part 1, pp. 127-142. (2014)
- [15] Smullen, D., Gillett, J., Heron, J., and Rahnamayan, S. "Genetic algorithm with self-adaptive mutation controlled by chromosome similarity," in: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 504-511. IEEE. (2014)
- [16] Srinivas, M., and Patnaik, L. M. "Adaptive probabilities of crossover and mutation in genetic algorithms," in: IEEE Transactions on Systems, Man and Cybernetics, 24(4), pp. 656-667. (1994)
- [17] Kozeny, V. "Genetic algorithms for credit scoring: Alternative fitness function performance comparison. Original Research Article Expert Systems with Applications," 42(6), 2998-3004. (2015)
- [18] Li, J., and Li, M. "An improved genetic algorithm for solving deceptive problems," in: 2005 IEEE International Conference on Granular Computing, Vol. 2, pp. 502-505. IEEE. (2005)