AN EFFICIENT AND ROBUST GENETIC ALGORITHM APPROACH FOR AUTOMATED MAP LABELING

Fan Hong[1]* Liu Kaijun[2] Zhang Zuxun[1]

(1 National Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing of Wuhan University, No.129 Luoyu Road, Wuhan, China, Zip code:430079)
(2 Management college of Huazhong University of Science and Technology, Luoyu Road, Zip code: 430070)

Keywords: genetic algorithm, label placement, optimal combination problem, mutate on conflict gene, comparison experiment

ABSTRACT:

This paper put forward an new solution, which adopted the genetic algorithm to obtain the global optimal solution (approximate) of automated label placement of point feature. In the paper, the basic thought and design framework of using genetic algorithm to solve point feature labelling was firstly introduced, then, some practical technique and new improved method during the experiment procedure of genetic algorithm adopted by the author were presented in detail. Finally, in order to prove the advantage of genetic algorithm, some experiment were conducted which compare the efficiency of genetic algorithm with hill climbing algorithm, simulated annealing, neural network, etc. The result of comparison experiment was given out, which has proved the superiority of genetic algorithm, especially proved the genetic algorithm is a kind of high-efficient, robust, all-purpose algorithm with well-expansibility, and is the most promising solution for automated map labelling.

I. INTRODUCTION

Label is an important component of a map, with the aid of map label user can recognize the important objects and obtain its relevant information of objects. However for a long time, map label has been a time-consuming manual work. So the automated map label has been always one of the important study contents of computer-aided cartography all the time. On the other hand, the automated map label is one artificial intelligence puzzle. It has been already proved that finding the optimal solution of automated map label is a NP-hard problem [Marks, 1991].

This paper put forward one new approach to find the global optimal solution of automated map labeling with genetic algorithm. Genetic algorithm is a kind of method with global searching characteristic. It originates from simulating the biological evolution course of nature, and possesses many merits including high-parallel, self-organization, self-adaptation, self-learning and high efficiency, etc. Besides, genetic algorithm is simple, easy to operate and expand. Through plenty of experiments, we think genetic algorithm is a potential solution for automated map labeling.

II. TRADITIONAL METHOD'S DISADVANTAGES

It has gone nearly thirty years since Yoeli began to study point feature labeling, and many scholars have put forward various method to solve point feature labeling problem [Yoeli, 1972; Imhof , 1975; Ahn Freeman, 1984; Langran and Poilker, 1986; Zoraster, 1991; Christensen ef.al, 1995, 1997]. In the eighties of 20th century, a lot of automated labeling expert system were developed [Zoraster, 1991] that includes NAMAX, AutoNap,etc. The disadvantages of expert systems is low-efficiency and great developing cost. Another routine algorithm is exhausted searching algorithm [Jones, 1989; Cook and Jones, 1990; Ebinge and Goulette, 1990; Doerschler and Freeman, 1992]. These algorithm can be expressed as searching in digraph or direction tree, which might cause concatenate backtracking and even deadlock, So its severest problem is low efficient, therefore it is only fit to small-scale problem. Another kind of important algorithm is a kind of searching strategy based on the problem solution space. Representative examples include the energy minimization algorithm put forward by Hiersch [Hiesh, 1982] and the discrete gradient descent algorithm [Christensen ef.al, 1995, 1997], etc. There possibly appear two kinds of problems in these local search algorithms. First, they do not accept degenerated solution, therefore unable to jump out the "local minimum" trap; second, it may fall into dead loop.

III. POINT LABELING RULES

Consider point labeling, according to the common labeling principles, combining Chinese topographic map plotting pattern and regulations, in the labeling experiment of the residence topographic map with the scale 1:250000, our research will focus on the following three important principles especially:

1. The candidate positions and their priority: generally the candidate position of point feature labeling has four kinds of situations including four-candidate-position, five-candidate-position, eight-candidate-position and n-candidate-position. Four-candidate-position, as indicated in figure 1(a), regard the right as first, top, left and bottom followed respectively in succession, these labeling positions can be marked with priorities from 0 to 3. Five-candidate-position, as indicated in figure 1(b), similarly regard the right as first, the next are top, left, bottom, right-vertical respectively in succession, these labeling positions are expressed with priorities from 0 to 4. Eight-candidate-position, as indicated in figure 1(c), regard the right as first, the next are top, left, bottom, right-top, left-top, left-bottom, right-bottom respectively in succession, these labeling positions are expressed with priorities from 0 to 7. The n-candidate-position is illustrated in figure 1(d).

2. Forbid conflict: the labels of point features can't overlap (conflict) with one another.

* fh@mail.liesmars.wtusm.edu.cn

3. Forbid and avoid overlap: Point label should not overlap the important linear feature of the same color such as railways and major roads etc, While overlap is unavoidable, efforts should be made to decrease it.

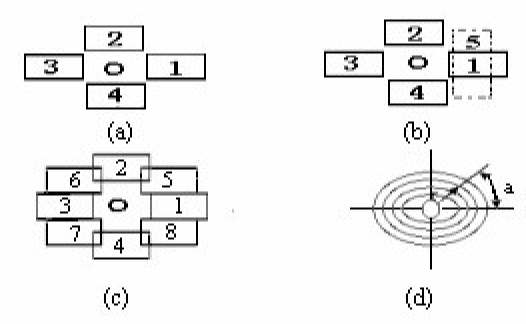

Figure 1 the candidate labeling position of point feature

This paper studies the labeling problem of the following two kinds of point features mainly. The algorithm of solving four-candidate-position is easy to expand to apply in the situations of 5, 8 and n candidate-position.

Problem I: consider the point feature in map, if adopting four-candidate-position labeling mode, namely choosing the right, top, left and bottom four candidate positions of point feature respectively (as indicated in figure 1(a), the distance between label and point residence is 1mm), try designing and implementing labeling algorithm to make globally conflict least.

In the actual labeling problem, besides considering eliminating label conflict, also need to consider dodging important map features and choosing the candidate point with higher-priority, thus the above problem becomes more complicated problem as follow.

Problem II: consider the point feature in map, if adopting four-candidate-position labeling mode, namely choose the right, top, left and bottom four candidate positions of point feature respectively (as indicated in figure 1(a), the distance between label and point residence is 1mm), try designing and implementing labeling algorithm, make globally conflict least, the overlap to other features minimum and the labeling position optimal.

IV. GENETIC ALGORITHM OF SOLVING POINT FEATURE LABELING

Genetic Algorithm was developed by Professor J.H.Holland, his fellows and his students in Michican University of U.S.A., in the sixties of the twentieth century. At present it has been applied to solve various optimal problems, such as layout scheme, self-adaptive control, game rules, pattern cognition, transportation problem, travelling salesman problem, optimal control and database query optimal, etc, most of them are famous NP-complete problems[Zhou Ming, 1999].

The biology has been always evolving according to the rule of "survival of the fittest" and natural genetics course, genetic algorithm is exactly the randomized calculating model originated from simulating the biological evolution course. In the solving course, genetic algorithm always keeps a population of potential solution. Begin from one initial population, through selection, crossover and mutating to produce the next generation population, in this way seek the optimal solution generation after generation until meeting the terminating condition. In order to solve one given problem, genetic algorithm must go through the following five steps generally [Zbigniew Michalewicz, 2000]:

(1) Determine the encoding framework;
(2) Generate initial population;
(3) Determine the fitness function;
(4) Design genetic operator, including selection, crossover and mutate operators;
(5) Determine the important parameters of genetic algorithm.

Because genetic algorithm is an all-purpose algorithm with extensive applicability, in design often need combine itself with the special rule of problem domain. In application course, we have put forward some optimization strategies according to the characteristics of labeling problem and has improved the performance of the algorithm greatly. The remains of this paper will in detail introduce our genetic algorithm and some crucial design theories and optimization strategies adopted by our genetic algorithm.

4.1 Determine encoding framework

The map label is the optimization target of genetic algorithm. One of the map placements can be expressed with a integer vector. Each component represents the localization of one label. Assumed there are m candidate positions, which can be expressed with codes of 0~m-1, for instance when considering four-position label, the four candidate positions can be expressed with the codes of 0~3.

Encode the map label placement with integer vector, and a piece of chromosome is a integer vector representing an instance of a map label placement. The length of chromosome is n (the number of point features label), and every component (gene) represents one point feature label, the domain of gene is [0, m-1], where m is the number of candidate positions, the gene code set of four-candidate-point labeling problem is {0, 1, 2, 3}. Now figure 2 give an example, it shows a map with twenty point features, one placements of this map and the corresponding chromosome encoding.

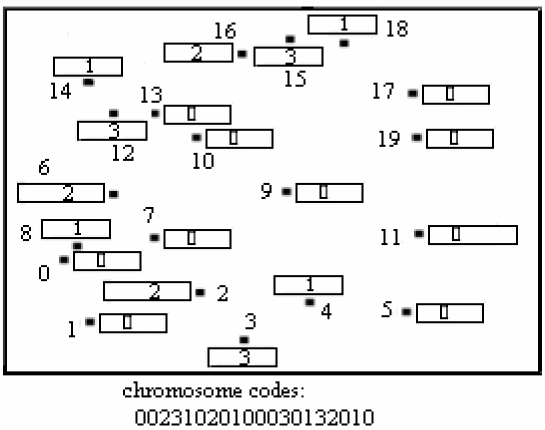

Figure 2 Point feature labeling placement and its encoding

4.2 Generate initial population

According to the characteristic of labeling problem, the following strategies to generate initial population was adopted.
(1) Randomly generate an initial population with certain size, and randomly select every gene for the chromosome.
(2) According to the characteristic of labeling problem, for all free labels (when selecting optimal position for them, overlap and conflict never appear), in initial population select the optimal positions for the corresponding genes of all chromosomes.

The above initial population strategy has selected the optimal positions for free labels. That means free label's position can be solved without through optimal process. It also means the reduction of the scale of problem and the acceleration of the evolvement of genetic algorithm.

4.3 Determine the fitness function

The target of labeling problem is to find the label placement with the highest quality. Therefore the fitness function is defined as the labeling quality evaluation function. Here adopt such a way: first define a labeling quality evaluation function which considers these factors including conflict, overlap, position priority and so on, now adopt the way of marking, namely mark the conflict, overlap and position priority of each label, and then calculate the total score of each label through weighted average of summation, finally, calculate the summation of all the labels, thus obtain the score of the whole labeling placement. The higher the score is, the higher the labeling quality is, this is exactly consistent with the meaning of the fitness function (the larger the fitness value is, the better the individual is). According to this thought, by referring to the demands of optimization target of different labeling problems, define the corresponding fitness function. Now introduce it with two examples of optimization targets.

1. The least conflict target

First consider the point labeling problem I; it only considers the optimization target of least conflict. Under this kind of situation, define of the fitness function see equation (1).

$$fit(L) = \sum_{i=1}^{N} E_{conflict}(L_i) \tag{1}$$

$$E_{conflict}(L_i) = \begin{cases} 1 & If(\forall i, 0 < j < n, j \neq i, d_{ll}(L_i, L_j) > 0) \\ 0 & otherwise \end{cases} \tag{2}$$

Where $E_{\text{conflict}}(L_i)$ is 0-1 conflict evaluation function whose definition is shown in expression (2), in which L_i represents the label of i-th point feature, when there is no conflict between L_i and the other labels the function equals 1, otherwise it equals 0. This kind of fitness function is defined as the sum of the labels which don't overlap with other labels. By using this kind of fitness function, genetic algorithm can solve the global conflict better.

2. The target of the least conflict, overlap and optimal position

Now consider the point labeling problem II, it has three optimization targets, and the fitness function needs to consider conflict, overlap and position priority, therefore define the fitness function as equation (3):

$$E(L_i) = E(L_{i,j}) = \begin{cases} W_{overlap}E_{overlap}(L_{i,j}, BF) & if\,(L_{i,j}..don't..Conflict) \\ +W_{position}E_{position}(L_{i,j}) & \\ 0 & otherwise \end{cases}$$

$$fit(L) = \sum_{i=1}^{N} E(L_i) \tag{3}$$

Where we let $W_{\text{overlay}} = 100; W_{\text{position}} = 1$.

The meaning of every symbol is as follows:

$E_{\text{overlay}}(L_{i,j}, BF)$ represents the overlap evaluation value when L_i is on the candidate position j, if adopting simple overlap evaluation function, it is defined as the highest importance weight of the features overlapped with the label, when there is no overlap, $E_{\text{overlay}}(L_{i,j}, BF) = 99$, the higher the importance of overlaid feature is, the severer the overlap is, accordingly the lower the overlap score is. BF_j represents the j-th background feature overlaid by the label; the predicate overlap(O_1, O_2) indicates the two objects O_1, O_2 overlap with each other. Now define $E_{\text{overlay}}(L_{i,j}, BF)$ as

$$E_{overlap}(L_i, BF) = \begin{cases} 99 & no...overlap \\ 99 - \max\{W(BF_i) \mid overlap(L_i, BF_i) \wedge BF_i \in BF\} & with..overlap \end{cases} \tag{4}$$

In equation (4), $W(BF_j)$ represents the importance evaluation function defined by background feature (it value called importance grade or weight). Similarly adopting the mark system 0~99, the score of the feature which can't be overlaid is 99, the lower the importance, the lower the score is, the score of the feature with the lowest importance is 0. Now define $W(BF_j)$ as equation (5):

$$W(BF_i) = \begin{cases} 0 & the\ \ minimal\ \ importance \\ 1 \sim 98 & the\ \ median\ \ importance \\ 99 & the\ \ maximal\ \ importance \end{cases} \tag{5}$$

$E_{\text{position}}(L_{i,j})$ represents the position evaluation value when L_i is on the candidate position j, adopt sorted position evaluation function, and is calculated according to equation (6). When the candidate positions are finite and can be enumerated (such as four-position labeling or eight-position labeling). We sort them in the descending order of their priority, let $Pos_j(L_i)$ represents the j-th labeling position of L_i, $Order(Pos_j(L_i))$ represents the order number of candidate position after sorting, we can define the position evaluation function as the difference of 99 and $Order(Pos_j(L_i))$. Namely the score of the position with the highest priority is 99, the scores of the other positions decrease in order. The definition of $E_{\text{position}}(L_{i,j})$ see equation (6):

$$E_{position}(L_i) = 99 - Order(Pos_j(L_i)) \tag{6}$$

By adopting this kind of fitness function, genetic algorithm not only solves conflict but also solve the optimization of the overlap and position priority.

In addition, if only consider the target of least conflict and most optimal position, let $W_{\text{overlay}} = 0$ in equation (3), then

We can have equation (7):

$$E(L_i) = E(L_{i,j}) = \begin{cases} W_{position} E_{position}(L_{i,j}) & If£¨L_{i,j}\ without\ conflict£© \\ 0 & otherwise \end{cases}$$

$$fit(L) = \sum_{i=1}^{N} E(L_i) \qquad (7)$$

Where $W_{position} = 1$.

Now consider the example in figure 2, the figure gives out its chromosome. Consider the targets of least conflict and most optimal position, adopt the fitness function (7), and calculate its fitness value:

$$fit(L) = \sum_{i=1}^{20} E(L_i) = 99\times10 + 98\times4 + 97\times3 + 96\times3 = 1961.0$$

4.4 design genetic operators

4.4.1 Selection operators

The labeling algorithm adopts roulette-wheel selection as selection method. When the population size is very large, can use elite strategy, namely retain the optimal individuals of the previous generation into the next generation's population. The implementation algorithm of roulette-wheel selection is as follow:

(1) Calculate the fitness value of individual $fit(V_i)(i = 1,2,...,n)$;

(2) Calculate the accumulative fitness value of individual $Accfit(V_i)(i = 1,2,...,n)$ and relative accumulative fitness value $\mathrm{Re}lAccfit(V_i)(i = 1,2,...,n)$.

(3) Generate a random r in [0, 1],

here suppose $\mathrm{Re}lAccfit(V_0) = 0$ If ,

$\mathrm{Re}lAccfit(V_{i-1}) < r \le \mathrm{Re}lAccfit(V_i) \qquad (i = 1,2,...n)$,

then select the individual i.

4.4.2 Crossover operator

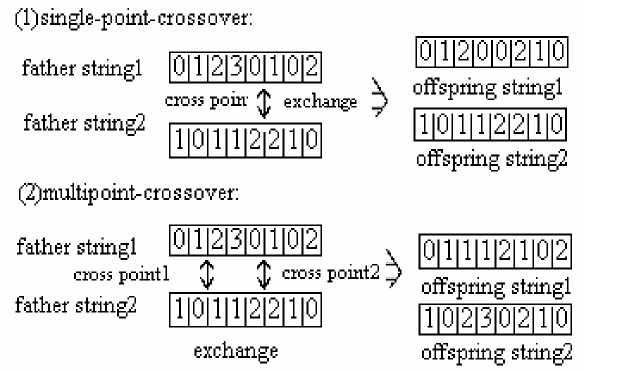

Figure 3 the example of point-crossover

The integer vector generally adopts two kinds of crossover operators: point-crossover and even-crossover. The experiments have proved there is no too much difference between the influences of two kinds of crossover operators on the performance of labeling algorithm. Therefore the labeling algorithm adopts the point-crossover strategy.

The point-crossover operator is divided into single-point-crossover and multipoint-crossover. The single-point-crossover randomly selects a cross point on two father strings and then exchanges the corresponding sub-strings of the two strings. The multipoint-crossover randomly generates several cross point every time, and then exchanges the corresponding sub-strings of father strings respectively [Pan Zhengjun, 1998]. Figure 3 gives a chromosome point-crossover example of two labeling placements with the string length 8 and the gene code set {0, 1, 2, 3}.

4.4.3 Conflict gene mutate (namely mutate on conflict gene)

For integer vector encoding, the common mutation includes point-mutation and even-mutation. The former selects single point, the latter selects point according to some template, and then randomly relocate the selected point.

As for map labeling, we put forward a new mutation operator which is called conflict gene mutation to replace the routine point-mutation and even-mutation. The basic theory of conflict gene mutation is that: select the gene of conflict label, randomly generate a labeling code to replace the original gene.

Experiments have proved the conflict gene mutation is very effective. Figure 4 shows the comparison result of the conflict gene mutation and even-mutation with the same genetic parameters (the number of iterations is 300, population size is 50, crossover probability is 0.75, mutation probability is 0.2). From figure 4 we can find that the conflict gene mutation is obviously superior to the even-mutation. The possible reasons are as follows:

(1) The point-mutation and even-mutation are blind, and the conflict gene mutation utilizes the heuristic information of label conflict to improve the bad sub-solution, so the probability of obtaining good sub-solution is bigger.

(2) In actual maps, there is little conflict in the area with sparse labeling point, the mutation probability should be smaller, however in the area with dense labeling point there is more conflict, and the mutation probability should be relatively larger. The conflict gene mutation meets this demand.

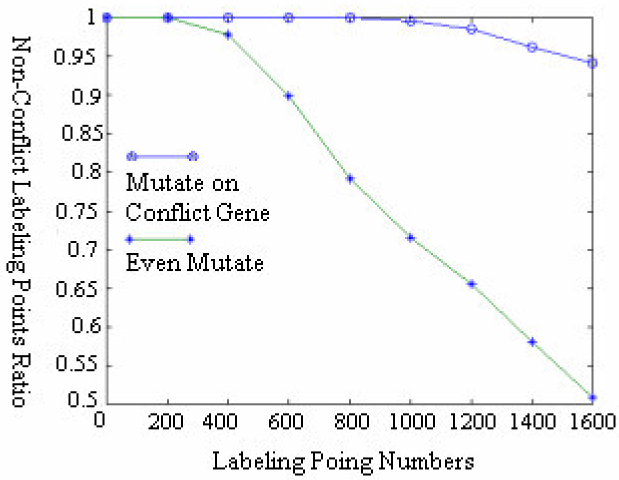

Figure 4 compare conflict gene mutation with even mutate

4.5 determine the important parameters of genetic algorithm

According to references and experiment conclusion, there are the following principles in determining the genetic parameters of automated labeling:

(1) The population size N: it affects the validity of genetic algorithm, recommended it no more than 300. In this paper the domain of N is [30, 300].
(2) Crossover probability P_c: it controls the frequency of crossover operation. Generally it is between 0.25 and 0.85. In this paper it is between 0.6 and 0.8.
(3) Mutation probability P_m: it is the second factor in increasing the diversity of population. Generally P_m is between 0.01 and 0.2. This paper doesn't have to use this parameter.
(4) Terminating number of generation: when population evolves over the specified largest evolution number of generation, terminate the evolvement course, so this parameter should guarantee the population has matured. There are two conditions to judge whether the population has matured: (1) through several operations, the approximate optimal individual can be gotten stably; (2) continue evolving, the optimal individual is not improved obviously again.

V. GENETIC ALGORITHM EXPERIMENTS AND CONCLUSION

The experiment of genetic algorithm scheme is composed with two parts of data. The first part is some random maps generated by the algorithm developed by ourselves; the second part is the actual topographic maps that includes three complete feature topographic maps from National Topographic Map Database of 1:250000, among them every map is composed with nineteen layers including hydrogen, road, vegetation, boundary, and so on. The three topographic maps contain point residences 2511, 1651 and 2734 respectively.

From the evolvement experiment on the map containing 50~3000 point features, we can find that in the iteration course, with the gradual increment of the fitness value of optimal individual, genetic algorithm becomes steady, this indicates the population has matured. The algorithm should be terminated after the population has matured. Usually the map with no more than 3000 point features becomes matured within 300 generations.

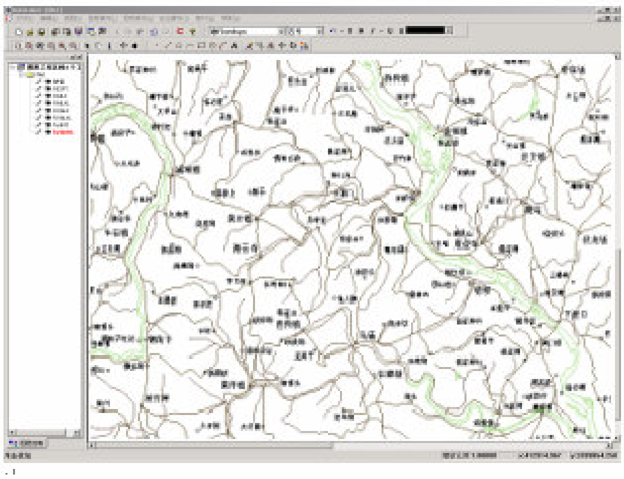

Figure 5 the labeling result of H4810 using genetic algorithm

The experiments also include the comparison of using genetic algorithm on maps with different complexities and the comparison of using genetic algorithm with different parameters on the same map. The experiments indicate that when processing the problems of different complexities with the same genetic parameters the results are different. When keeping certain population size, the increment of evolvement generations will increase the fitness of solution. But once reach certain degree, when the population is matured, the solution will not be made better. If the evolvement generations are very few, the simple increment of population size is not very useful to improve the solution.

The experiments include the experiment on three actual topographic maps which belong to the problems of moderate difficulty. Genetic algorithm can eliminate nearly all the conflicts (only 1 or 2 are not eliminated) in 20~30 seconds. Figure 5 is a part of the labeling result of H4810 (2511 point features).

VI COMPARISON EXPERIMENT AND CONCLUSION

In order to verify the performance of genetic algorithm introduced in this paper, we compare genetic algorithm with simple random algorithm, hill climbing algorithm, simulated annealing and neural network algorithm.

The simple random algorithm is one of the simplest algorithms. It specifies randomly a labeling position for every point feature, but not implement any optimization, it's algorithm quality is the lowest limit of labeling quality.

Hill climbing algorithm is a kind of simple local optimization algorithms. It begins from an initial solution of randomly given labeling placement (or be calculated by other methods or be specified directly), search the n adjacent new solutions generated randomly from the current solution, and select the optimal solution and continue searching in new solutions until the solution can't be improved again. In order to guarantee the comparability, we set up the numbers of iterations as 200~300 generations in experiments, namely let the program start to search in 200~30 different initial solutions, and search thirty adjacent strings every iteration.

Christensen and his fellows have put forward a kind of point feature labeling algorithm based on simulated annealing. Simulated annealing algorithm is a kind of simple global searching algorithm, and it is the improved hill climbing algorithm, which adopts randomly relocating in labeling, but allows the "degenerated solution" with certain probability in order to jump out the local minimum. About the kernel algorithm please refer to [Christensen, 1995]. Christensen has proved simulated annealing algorithm possesss a lot of performance superior to traditional algorithms. The simulated annealing in this paper adopts the processing flow and parameters in Christensen's work. In order to guarantee the comparability, we set up the number of iterations as 200~300.

Neural network algorithm adopts the model put forward by Fan Hong and Zhang Zuxun [Fan Hong, Zhang Zuxun, 1997], after setting up the neural network of solving point labeling problem, let the network run iteratively, the running result is regarded as the labeling placement. In order to guarantee the comparability, we also set up the number of iterations as 200~300.

The experimented data is a group of point maps generated randomly. The experiment methods are carried on based on the same data background and subsidiary data conditions. Before using algorithms we have set up the same conflict detection table and overlap detection table.

The experiment compares the labeling qualities of different algorithms mainly. In order to compare conveniently, all consider the four-position labeling problem of point feature, and consider the following two optimization targets: (1) only consider conflict optimization; (2) consider conflict and position optimization, but not consider overlap.

Under the situation of only considering conflict, we expressed the labeling quality with the ratio of non-conflict labels to the total of labels. We randomly generate 8*5 map sheets of point feature with 200, 400, 600, 800, 1000, 1200, 1400, 1600 point features respectively, figure 6 shows the comparison result of these algorithms. The result in figure 6 is the average of the labeling results on five maps.

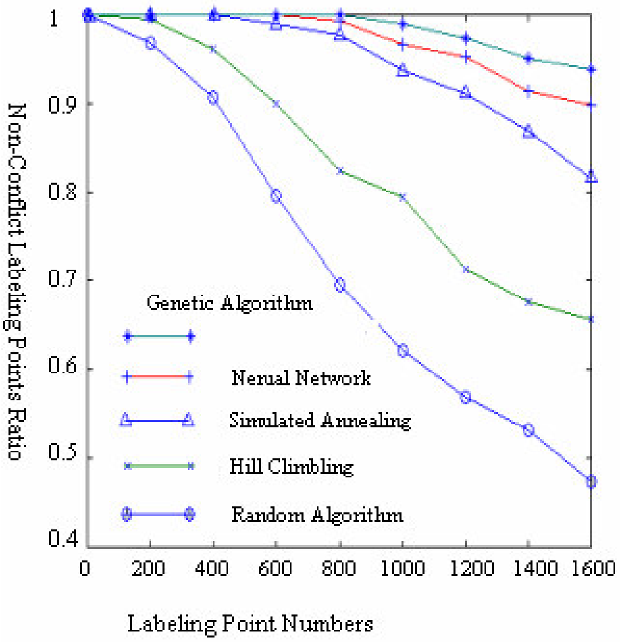

Figure 6 performance comparison among 5 kinds of algorithms

From the above comparison experiment we can draw two conclusions.

(1) From figure 6 we find that the solution of genetic algorithm has the highest quality to the map with the same complexity, the next is neural network algorithm, the next again is simulated annealing algorithm and hill climbing algorithm, and the quality of random algorithm is the lowest, whose labeling quality is the lowest limit of available solution quality. From the angle of labeling quality, genetic algorithm> neural network algorithm > simulated annealing > hill climbing algorithm. Genetic algorithm has the highest comprehensive performance.

(2) Genetic algorithm introduced in this paper is a kind of robust and expansible automated labeling algorithm with well-performance. It possesses the following merits: easy to add the consideration of other optimization factors, well expansibility. The encoding form can be determined by problem, and easy to expand according to problem. In addition genetic algorithm is very robust, it will not generate invalid solution. The parameters of genetic algorithm are easy to modulate. Its primary parameters have been determined by system, the workload of parameter modulating is very little.

ACKNOWLEDGEMENTS

Thanks for the supporting from Natural Science Fund of P.R.China (No. 40001019).

References:

Zbigniew Michalewicz, 2000, evolutionary programming, the press of science.

PanZhengjun, KangLishang, ChenYuping,1998,evolutionary calculation, the press of tsinghua Univ.

Zhou Ming, Sun Shudong, the principle and application of Genetic Algorithm, the press of National defense industry.

Herbert Freeman and John Ahn. On the problem of placing names in a geographic map. Int. J. of Pattern Recog. and Art. Intell., 1(1):121-140, 1987.

Hiersch,S.A.,"An Algorithm for Automatic Name Placement Around Point Data.", 1982, The American Cartographer, vol 9(1)

Imhof, E., 1975, "Positioning Names on Maps",The American Cartographer, vol 2(2), pp 128-144

James E. Mower, 1993 ,"Automated Feature and Name Placement On Parallel Computers", Cartography and Geographic Information Systems, Vol.20(2), pp. 69-82

Jefrey S. Doerschler and H. Freeman. A rule-based system for dense-map name placement. Comm.of the ACM, 35:68-79, 1992.

Anthony C. Cook and Christopher B. Jones. A Prolog rule-based system for cartographic name placement. Computer Graphics Forum, 9(2):109-126, 1990.

Jon Christensen, Joe Marks, and Stuart Shieber. An empirical study of algorithms for point-feature label placement. ACM Transactions on Graphics, 14(3):203-232, 1995.

Lee R.Ebinger and M. Goulette,1990,"Noninteractive Automated Names Placement for the 1990 Decennial Census.", Cartography and Geographic Information Systems, Vol. 17(1) , pp. 69-78

Yoeli,P.,1972, "The logic of Automated Map Lettering",The cartographical Journal, vol. 9(2),pp.99-108

Christopher B. Jones and Anthony C. Cook. Rule-based name placement with Prolog. In Proc. Auto-Carto 9, pp 231-240, 1989.

David S. Johnson, Umit Basoglu.,1989,"The Use of Artificial Intelligence in The Automated Placementof Cartogrphic Names", Proceedings of Auto-Carto 9

Doerschler,J.,and H. Freeman,1989,"An Expert System for Dense-Map Name Placement." Proceedings of Auto-Carto 9,pp. 215-224

Freeman,H.,J.Ahn.,1984,"AUTOMAP:An Expert System for Automatic Map Name Placement" Proceedings of the First International Symposium on Spatial data Handling, pp 544-571

David S. Johnson, Umit Basoglu.,1989,"The Use of Artificial Intelligence in The Automated Placementof Cartogrphic Names", Proceedings of Auto-Carto 9

Doerschler,J.,and H. Freeman,1989,"An Expert System for Dense-Map Name Placement." Proceedings of Auto-Carto 9,pp. 215-224

John Ahn and Herbert Freeman. AUTONAP - an expert system for automatic map name placement. In Proceedings International Symposium on Spatial Data Handling. pages 544-569, 1984