

A Genetic Algorithm for Solving Multistage Graph Problem

Jinqing Tian , Xiaofeng Wang, Hongsheng Ding

College of Computer Science, North University for Nationalities, Yinchuan 750021, China

E-mail: 522596431@qq.com

Abstract: A multistage graph is center problem of computer science, many coordination and consistency problems can be convert into multistage graph problem. We obtained the fitness function by coding the vertex of multistage graph, and designed the genetic algorithm for solving multistage graph problem. Experiment results show that this algorithm is very effective and feasible.

Keywords: genetic algorithms; multistage graph problem; algorithm design

1 INTRODUCTION

A multistage graph is center problem of computer science [1]. Many coordination or consistency problems can be convert into multistage graph problems for solving, such as engineering technology, artificial intelligence, military, concurrency control, transportation and other fields. The key problem of multistage graph is to calculate the shortest path between two points in the graph which is a classical problem in graph theory. At present, there are many methods for solving the shortest path, such as Bellman-Ford algorithm, Dijkstra algorithm, Floyd-Warshall algorithm linear programming algorithm, ant colony algorithm.

For the shortest path problems, the Dijkstra algorithm and Floyd algorithm are excellent algorithm. However, the large scale directed graph is the research focus; many scholars have brought up the parallel algorithms for the problems. For example, a parallel Dijkstra algorithm is proposed in [2-4], which is realized by using the multi-thread method. But it's serial execution essentially. According to the distance matrix and path matrix, the parallel Floyd algorithm is proposed in [5, 6], which is still a serial algorithm due to the limitation of the processor. Therefore, if we do not consider the multi-processor factor, all parallel algorithms are still not implemented in parallel computing in a real sense. In order to obtain the exact solution of the problems, it is bound to increase the running time of the algorithm. In fact, it can meet engineering application so long as the solution can achieve a certain accuracy, which usually doesn't require a precise solution. That makes the uncertain algorithm is widely used to solve the problems. For example, [7] is used to solve the problems by using the ant colony algorithm. And another method is based on water channel principle [8]. All these methods have their shortcomings, such as easy to fall into

local optimum; the algorithm has a long convergence time; and is not adapted to multi-objective problem, etc.

In this paper, a genetic algorithm is designed to solve the multistage graph problem, and the fitness function model is given by encoding. Using filtering technology to abandon the individual which is far away from the target, and then achieve the rapid convergence. Experimental results show that the proposed method is very effective and feasible.

2 BASIC KNOWLEDGE

Multistage Graph $G = (V, E)$ is a directed graph. It has the following characteristics: the nodes in the graph are partitioned into $k \geq 2$ disjoint sets V_i . V_1 and V_k are only one vertex s (source) and one vertex t (sink). All of the edges in the graph have the following properties: if $u \in V_i$, then $v \in V_{i+1}, 1 \leq i < k-1$, and each side $\langle u, v \rangle$ are accompanied by cost $c(u, v)$. A path from s to t is the sum of the cost of this path.

Multistage graph problem (MPG) is to calculate the minimum cost path from s to t . Each set V_i denotes nodes set of a stage of diagram. Because of the constraints E , the path from s to t is beginning from the first stage and the end the k stage [1]. Figure 1 shows a 6 stages graph.

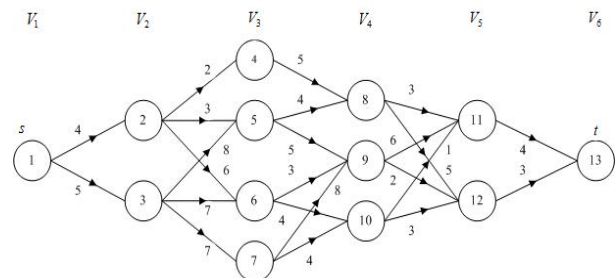


Fig 1 A 6 stages graph

Many practical problems can be represented by multistage graph. For example, resource allocation problem, that is the assignment of N resources to R projects; the 0/1 knapsack problems, network data transmission, etc. These can be transformed into MPG to solve. That is to say, the practical problems of multistage decision-making can be solved by converting into MPG. Because it satisfies the optimality

This work is supported by National Nature Science Foundation of China(NO.61462001), Ningxia Colleges Projects Research Foundation (NO.NGY2013093), Agricultural IOT Innovation Team Project Foundation of North University for Nationalities. North University for Nationalities Projects Research Foundation(2012Y040).

principle, the MPG can be solved by using the dynamic programming method to solve.

In the field of artificial intelligence, the core idea of Genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems. GA is a family of computational models inspired by evolution algorithm (EA) [7], which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. GA optimizing process has always maintained the whole population evolution. Therefore, GA has opened up a new way to solve many difficult problems of global optimization problems [7]. In this paper, the genetic algorithm is proposed to solve the MGP. The method provides a new way to solve this kind of problem.

3 GENETIC ALGORITHM DESIGN FOR MGP

3.1 Encoding method

Encoding is the first problem and crucial step in the application of genetic algorithm [9], and using the order of nodes as the encoding way generally in the traditional GA. However, due to the stage characteristics of the MGP: if we use traditional encoding way, there will be a lot of redundant encoding in stage i (path nodes $v \in V_i, 1 \leq i \leq k$), which will increase the algorithm's running time. According to this problem, we designed a simple and effective method with the stage binary encoding. For a chosen path $s, v_2, v_3, \dots, v_{k-1}, t$, it is clear that the source s and sink t do not need encoding. The number of nodes in state set V_i is $|V_i|$, the gene length is l_i corresponding to the nodes $v \in V_i, 2 \leq i \leq k-1$, and it should meet $2^{l_i} \geq |V_i|$. e.g. If the size of the set V_i is 3, then $l_i = 2$. The chromosome length of the entire path is L .

$$L = \sum_{i=2}^{k-1} l_i \quad (1)$$

The gene corresponding to nodes $v \in V_i, 2 \leq i \leq k-1$ is the position sequence (starting from 0). As shown in Figure 1, one of the paths is $1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 12 \rightarrow 13$. With this encoding way, $|V_2| = 2, l_2 = 1$, and the encoding of node 2 is 0; other stages of the gene encoding similar to the above. The encoding of the path as figure2. Then the encoding length of entire path is $L = 6$.

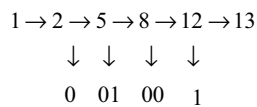


Fig 2 Gene coding of a path

There are two problems with this encoding mode, and the method of dealing with the problem is given as follows:

1) Encoding redundant: if $|V_i|, 2 \leq i \leq k-1$ is not a power of 2, there is a redundant of encoding. For example, $|V_4| = 3, l_4 = 2$ the two bit encoding has four states $\{00, 01, 10, 11\}$, but $|V_4|$ only three nodes, the encoding 11 is redundant, if the encoding 11 appears in the process, we

can use formula $Temp(11)\%|V_4|=0$, $Temp(11)$ is the corresponding decimal of the binary encoding, which the result is converted into the corresponding binary encoding 00.

2) Bad path solving: If the edge $\langle v_i, v_{i+1} \rangle, 1 \leq i \leq k-1$ does not exist, but in the process of GA appears this encoding, called the path for the bad path. Because the probability of a bad path is relatively small, we can delete the bad path directly. We define the bad chromosome set in the actual algorithm. If the chromosome is matched with the elements of B, then it is a bad chromosome. For example, the node 2 (encoding 0) and the node 7 (encoding 11) are disconnect in Figure 1, the encoding is 011*** (* indicates 0 or 1), which is a non-existing edge corresponding to the node 2 and node 7. we can use fuzzy matching method to search for the bad chromosome, which is easy to know that the number of elements in set B is the number of edges $\langle v_i, v_{i+1} \rangle, 1 \leq i \leq k-1$ not in multistage graph.

3.2 fitness function

MGP is solving the minimum cost path from the source s to the sink t . The basic fitness function is defined as (2):

$$F = -\sum_{i=1}^{k-1} c(v_i, v_{i+1}) + C_{\max} \quad (2)$$

$c(v_i, v_{i+1})$ denotes the cost between v_i and v_{i+1} , the C_{\max} guarantees F for the positive integer. We can get the appropriate fitness for a path (also called chromosome). But the real price needs to minus C_{\max} and then invert.

3.3 selection operator

The selection is an operation of the fittest survival in GA, which determines the search direction. One of the main factors affect the genetic search is the diversity of the population [10]. Generally, the individual is selected and genetic to the next generation population, which is related to the fitness of the individual. The more fitness is, the more probability of being selected is, and vice versa. However, there are some individuals have little chance to reproduce themselves due to their small fitness proportion, and they are premature elimination easily. Thus, the diversity of the whole group is not guaranteed. In order to guarantee the diversity of the whole population, the individual with the least fitness is kept every time.

3.4 crossover operator

According to the cross probability P_m , a pair of paternal chromosome are chosen to generate a random number $t, 1 < t < L$, and the cross operation is to do at the location t of the two chromosomes.

3.5 mutation operator

Mutation is the process of generating new individuals instead of the selected genes. According to the given mutation probability P_c , the chromosome is selected to generate the mutation, and the mutation gene is generated

by random number method, then the corresponding genes are taken back.

3.6 Algorithm stopping condition design

If the evolution has reached the request of algebra, or the optimal chromosome is no longer improved in the M generation, then the algorithm is terminated.

The algorithm for solving the problem of multi graph is described as follows:

Step1: Set various parameters, including population size N , gene block length array $L[k]$, crossover probability P_m , mutation probability P_c , reproductive number T , bad chromosome collection B , evolution algebra M , and so on.

Step2: Using the method of generated chromosomes and initial population with random. We should redo to get good population if there are bad chromosomes.

Step3: Using the operator of selection, crossover and mutation to obtain the next generation of population, if there are some bad chromosomes in the population, it must be deleted, and regenerating randomly qualified chromosomes, to ensure that the size of the population.

Step4: Stopping the algorithm execution if satisfy the conditions to stop; otherwise jump to *Step3*.

The algorithm we proposed is different from the basic genetic algorithm where the selection operator and the stop conditions, as well as the processing of the bad chromosomes because the encoding mode in stage. When the selection is performed, the individual with a minimum fitness is reserved to ensure the diversity of the population. This process will not increase the complexity than the basic genetic algorithm.

Except the basic GA stopping conditions, we add a condition that the optimal chromosome is no longer improved in the continuous M generations. It only needs to compare once a time, and a storage space to store the optimal chromosome for each breeding. In the case that has found the best and did not reach the number of breeding, it can reduce the running time of the algorithm, and it does not increase the time complexity of the algorithm.

Set $t = |B|$, t is the number of elements in B . According to previous definition, the number of elements in B is the number of edges $\langle v_i, v_{i+1} \rangle, 1 \leq i \leq k-1$, which is none-existence in multistage graph. Set checked time for one chromosome to be c (constant), the whole algorithm execution time of checking bad chromosomes must be less than $Ntc(T+1)$. So the complexity of this algorithm is mainly determined by the basic genetic algorithm, but it is difficult to give an accurate mathematical analysis of the GA performance. The GA complexity is mainly determined by the size of the problem, genetic operators, and the convergence degree of the genetic algorithm itself.

4 NUMERICAL EXPERIMENTS AND ANALYSIS

4.1 parameter control

In the process of the implementation of the algorithm, the choice of parameters can play a very important role in the process of finding a satisfactory solution. The main parameters are cross probability P_m , mutation probability P_c , population size N , reproductive number T , and so on. In this paper, we set P_m and P_c with the basic genetic algorithm's parameters, $P_m = 0.05, P_c = 0.80$. The small values of N may lead to premature convergence of the algorithm, and the larger one can reduce the algorithm speed, the values are commonly to take between 20 and 50. T is a parameter that indicates the number of reproductive times of genetic algorithm control, generally T is to take between 100 and 2000. We Set $T = 100$ in our algorithm. If the value of M is too small, the solution is not necessarily optimal; otherwise, it can not work in reducing the reproductive times, so we set $M = 10$ according to the experimental. It must be selected the appropriate parameters in the actual calculation, according to the specific problem.

4.2 algorithm verification

In order to test the practical application effect of the algorithm to the multistage graph, the algorithm is verified in this paper. The platform for experimental: CPU is 2.40GHz P4, the memory is 512MBytes, the operating system is XP win, and the debugging tool is turboc2. The validation results are shown in table 1.

Using GA to solve the problem of multi graph is a good method. While it has not show more superiority in many simply problems, it has advantages in dealing with the complex MGP. Because of the limitation of M , the population m don't propagate to the maximum reproductive number, and the optimal solution is obtained, which shows that the running time of the algorithm do reduce because of the improvement of the stopping conditions.

Table1. Validation results

K stage graph	Nodes number	Breeding times	found the optimal solution?
10	33	46	yes
15	49	57	yes
20	87	48	yes

5 CONCLUSIONS

In this paper, we propose a method for MGP, and improve the selection operator and the stopping conditions of the algorithm. Through the experimental verification, this approach is very good to solve the problem of multistage graph. In fact, many practical problems can be represented by a number of graphs, such as the problem of resource allocation, knapsack problem, etc. The further work of this paper is to design the simulated annealing algorithm to

solve the problem of MGP, and analyze the complexity of the algorithm.

REFERENCES

- [1] Yu Xiangxuan. Computer algorithm basis, Huazhong University of Science and Technology press, Wuhan, China, 2006. D. Z. Cheng, Matrix and Polynomial Approach to Dynamic Control Systems, Science Press, Beijing, China, 2002.
- [2] Lu Zhao, Shi Jun. Design and implementation of parallel shortest path search algorithm, computer engineering and applications, 2010, 46 (3): 69-71.
- [3] Shao Huizu, Based on Java multi-threading realization during all apexes most short- path parallel algorithm, Micro computer information, 2007, 23 (7): 126236-237.
- [4] Kamesh Madduri. David A. Bader, Jonathan W. Berry, Joseph R. Crobak, An experimental study of a parallel shortest path algorithm for solving large-scale graph instances, ALENEX '07, New Orleans, LA, USA, January 6, 2007.
- [5] Sun Shixin, Zhou Yimin, Tian Ling. A practical all point to the shortest path parallel algorithm for . Computer Applications, 2005, 25 (12): 2921-2923.
- [6] Hua Wang, Ling Tian. A practical parallel program algorithm for all pair shortest path based on pipeling. Journal of Electronic Science and Technology of China, 2008,6(3): 329-333.
- [7] Darrel Whitley, A genetic algorithm tutorial,, Stat&tics and Computing (1994) 4, 65-85.
- [8] Cheng Sanqin, Another Solution to Multistage Graph Problem Based on Water Channel Principium, Software Guide, 2005, (18) : 42-43.
- [9] Saber M. Elsayed, A new genetic algorithm for solving optimization problems, Volume 27, January 2014, Pages 57-69.
- [10] Whitley D. The GENITOR algorithm and selection pressure: why rank based allocation of reproductive trials is best, Proceedings of the Third International Conference on Genetic Algorithms, San Mateo: Morgan Kaufmann Publishers,1989:116-121.