A New Crossover Algebra of GA for Solving the Degree Constrained Minimum Spanning Tree Problems

Hui Li, Kai Shi, Huanran Li, Sheng Lin, Guangping Xu, Shuangxi Li
Tianjin University of Technology
2966788181@qq.com, shikai0229@163.com, 164940886@qq.com, {lins, gpxu, sxli}@tjut.edu.cn

Abstract. The degree constrained minimum spanning tree (DCMST) problem seeks a spanning tree of a connected graph with the smallest weight. It is of crucial importance to the design of communication networks. There is no polynomial algorithm for solving DCMST problem, heuristic algorithm is usually used to find an approximate solution. The heuristic algorithms, such as genetic algorithm (GA), face the problems on tree heritage, which means the offspring may not be a feasible solution. To address this problem, we propose a novel crossover algebra which considers both the edge learning and structure learning, and can guarantee the offspring be a feasible solution. Experiment results validate the effectiveness of the proposed algorithm.

1. Introduction
Degree constrained minimum spanning tree (DCMST) is the problem about generating a minimum spanning tree with degree constraint. It’s closely related to the problem about network design and network optimization. DCMST is widely used in computer network, communication network, transportation network and other network related problems. For example, with the increasing cost of infrastructure and equipment, the construction of today’s communication network and transport network impose more requirements and restrictions on the structure or function of nodes. Most of the nodes have limited storage and transfer ability, so it can connect to a finite number of nodes. This is the application of degree constrained minimum spanning tree in real life.

It has been proved that DCMST is NP-Hard, a heuristic algorithm is usually used to search for the best solution within acceptable computational cost [1]. The traditional heuristic algorithm can find the minimum spanning tree quickly when the edge weight is taken into account in the search process. But it is easy to fall into the local optima. So there is no guarantee to find an effective solution for DCMST [2]. Tradition heuristic algorithm faces the problems of tree representation and new solution generation which degrades its performance for finding good results.

A good tree representation method should have the following properties: it can represent all possible trees; it is unbiased in the sense that all trees are equally; it can represent only trees; it can be easy to go back and forth between the trees and codes; it should be localized in this sense that small changes in the representation make small changes in the tree. Tree representation should cover the entire space of solutions, produces only feasible offspring, and possesses locality, and all these necessary characteristics, in order to use genetic algorithm effectively [4]. However, for DCMST, the tree represents still faces some problems: The offspring are not necessarily represented in the form of trees, and the gap between them is too large. Aiming at addressing this problem, in this paper, we propose a novel crossover algebra to improve the feasibility and locality of GA algorithm, thus obtaining better results for DCMST problem.
2. Related work
The problem of the degree constrained minimum spanning tree is a well known problem. There are many algorithms proposed to solve the problem. The following will briefly introduce the related research in recent years.

The first, a branch and bound algorithm is proposed by Narula and Ho [5]. The step to select next node is rigidly and blindly, and it gives no priority to node that may quickly retrieve the answer node. The minimum spanning tree problem with degree constraint can be expressed as a linear 0-1 integer programming problem, which is solved by the construction method based on the original heuristic and dual heuristic and the branch and bound algorithm [6].

Secondly, a new representation method and an appropriate mutation operator can be used to solve the degree constrained minimum spanning tree problem. As the representation of candidate solutions and mutation operators are the basic design choices of evolutionary algorithms, this method has better locality and efficiency performance [7].

In addition, Prufer number can also be used as the chromosome in the current study to improve the crossover and mutation process of the existing genetic algorithm, so as to obtain high locality, heritability and self-adaptation performance [8].

In summary, most of the optimization improvements are based on crossover process. So they have the problem for generating a feasible solution in the neighborhood. Therefore, in this paper, we will propose a new crossover algorithm to solve these problems.

3. Crossover operator
As described above, the crossover operator, which is crucial to the performance of GA, faces some problems on generating new solutions. This means that the new solution may lose the character of feasible and locality. In this part, we will analysis these problems further and then design a new crossover operator to solve these problems.

3.1. Problems of traditional crossover algorithm

3.1.1. Crossover algorithm. In the original crossover algorithm, a pair of parents is selected for generating new children. In this process, a cross point k will be set at first, then the two parents exchange the genes from the cross point. Let the chromosomes of the parents be \( a(1,2,\ldots,a_n) \) and \( b(1,2,\ldots,b_n) \), and the cross point is k. After the crossover process, the new children are
\[ a(1,2,\ldots,a_{k-1},b_k,b_{k+1},\ldots,b_n) \] and \( b(1,2,\ldots,b_{k-1},a_k,a_{k+1},\ldots,a_n) \) respectively.

3.1.2. Problems. In this part, we will show the problems of traditional crossover algorithms under different tree representation method.

- Unfeasibility. As described above, one new generated child after the crossover process is \( (a_1,a_2,\ldots,a_{k-1},b_k,b_{k+1},\ldots,b_n) \). However, in some representation methods, the new child may not be a tree. Although the Prufer number representation method can guarantee that the new child is a tree, it may violated the node degree constraint.

- Locality. The new generation should keep the similarity of the apparent. However, the Prufer number method, which has the advantage of feasibility, will lost the character of locality at the same time.

3.2. A Novel Crossover operator

3.2.1. Basic idea. The traditional genetic algorithm only considers the edge heritage, which means the child inherits one half edges of the parent. In our mechanism, we propose a novel idea of structure heritage. Structure heritage means that the child will inherit the structure character of the parents. So in the new crossover mechanism, the children will inherit both the edge and structure character. It means
that a sub-tree of a child is isomorphic to one of its parents. Based on such idea, we will design our new crossover method in detail which incorporates both the edge heritage and structure heritage. The new algorithm comprises two parts: determining the cross point and generating new child trees.

3.2.2. Determining the cross point. The first step is finding two sub-trees in parents A and B respectively which has the same size of K. The size of K is about n/2. Let the nodes of the sub-trees be \((a_1, a_2, \ldots, a_k)\) of Parent A and \((b_1, b_2, \ldots, b_k)\) in Parent B.

3.2.3. Generating new children trees. Let the two parents be Parent A and Parent B and the two children be ChildL and ChildR. We will introduce the method to generate ChildL at first. The new generated ChildL comprises two sub-trees. One of which is from Parent A excludes the nodes \((a_1, a_2, \ldots, a_k)\), the other is a sub-tree with nodes \((a_1, a_2, \ldots, a_k)\) constructed based on the structure of sub-tree \((b_1, b_2, \ldots, b_k)\). The construct process can be describe as follows:

- If there is an edge between \(b_{ip}\) and \(b_{iq}\) (p, q={1…k}) in Parent B, then construct an edge between \(x_a\) and \(y_a\) for the ChildL. Otherwise, delete the edge between \(a_{ip}\) and \(a_{iq}\).
- Connect the two sub-trees, then forms the new ChildL.
- Similarly, we can generate ChildR which comprise one sub-tree from B excludes \((B_1, B_2, \ldots, B_k)\) and the other sub-tree \((B_1, B_2, \ldots, B_k)\) which is with isostructural to \((a_1, a_2, \ldots, a_k)\). Fig. 1 and Fig.2 shows an example the crossover process.

In short, we can conclude that the generating process is keeping one half of the structure of a parent unchanged, while updating the other half by exchanging the structure with another parent.
4. Genetic algorithm

Genetic algorithm (GA), also known as evolution algorithm, is a heuristic search algorithm, which is inspired by Darwin's theory of evolution and used for reference in the process of biological evolution. Understanding genetic algorithms requires understanding the concepts of populations, individuals, genes, chromosomes, heredity and mutation, and the concept of natural selection and survival of the fittest.

Population: the evolution of organisms takes place in the form of groups. Such a group is called a population. Individual: a single organism that makes up a population. Gene: a genetic factor. Chromosome: it contains a set of genes. Heredity and mutation: the new individual inherits the genes from each part of the parent, and at the same time has a certain probability of genetic variation. Natural selection and survival of the fittest: individuals who are highly adaptive to the environment are more likely to participate in reproduction, so more and more offspring will be born. In contrast, low-adaptability individuals are less likely to participate in reproduction, and fewer offspring [9].

There are three basic steps in genetic algorithm: selection, crossover and mutation. By genetic algorithm, the solution with low fitness function value is eliminated and the solution with high fitness function value is increased, so that the excellent gene can be preserved. In this way, it is possible to evolve individuals with high fitness function after the evolution of N generation.

5. Performance Evaluation

Each figure should have a brief caption describing it and, if necessary, a key to interpret the various lines and symbols on the figure. We have described the algorithms in detail before, and now we need to use benchmark data sets (CRD) to evaluate their performance.

In CRD data-set, each node corresponds to the horizontal and vertical coordinates of a point. Taking a 30-nodes data set as an example, the distance of each two nodes (the edge weight) needs to be calculated according to the Euclidean distance of the coordinates [10]. Therefore, we can get a 30 × 30 weight graph after the calculation, the same diagonal value of the horizontal coordinate is infinitely large, which means the node can not be connected with itself. The parameters of GA algorithm are set as Table1.

| The crossover probability | 0.2 |
|---------------------------|-----|
| Population                | 200 |
| Genetic generation        | 4000|
| Degree-constrained        | 3   |

Therefore, at a certain number of nodes, the crossover probability of 0.2, 200 population, genetic generation 4000, degree of 3 is limited. The results are shown in Table2 and Table3. From the table, we can see that the proposed mechanism can improve the performance by reducing the weight about 20 percent.

| 30 nodes | the original GA algorithm | the proposed algorithm |
|----------|---------------------------|------------------------|
| CRD300   | 10584                     | 9798                   |
| CRD301   | 11227                     | 8990                   |
| CRD302   | 11868                     | 9487                   |
| CRD303   | 12421                     | 9887                   |
Table 3. Target value of different algorithms at 50 nodes

|      | 50 nodes | the original GA algorithm | the proposed algorithm |
|------|----------|---------------------------|-----------------------|
| CRD304 | 13039     | 10289                     |                       |
| CRD305 | 11745     | 10267                     |                       |
| CRD306 | 12413     | 9386                      |                       |
| CRD307 | 13570     | 10086                     |                       |
| CRD308 | 11754     | 9154                      |                       |
| CRD309 | 11542     | 10384                     |                       |
| CRD500 | 22530     | 18919                     |                       |
| CRD501 | 17875     | 17368                     |                       |
| CRD502 | 21536     | 18130                     |                       |
| CRD503 | 19484     | 18058                     |                       |
| CRD504 | 20608     | 17448                     |                       |
| CRD505 | 20721     | 17703                     |                       |
| CRD506 | 21946     | 18146                     |                       |
| CRD507 | 21023     | 18062                     |                       |
| CRD508 | 21125     | 17486                     |                       |
| CRD509 | 21222     | 18063                     |                       |

6. Conclusion
In this paper, we investigated the DCMST problem using genetic algorithm. A new crossover algebra was designed for improving the feasibility and locality performance. Experiments results on CRD data-set showed that the proposed method can get better performance. In our future work, we will try to give a theoretical analysis of the new crossover algorithm.

Acknowledgment
This work was partly supported by an NSFC project under Grant No.11426163, Tianjin Nature Science Foundation Projects under Grant Nos.15JCQNJC0500, 17JCYBJC15600, a Tianjin Major Scientific and Technological Research Plan and Program under Grant No.16ZXHLSF00160.

References
[1] Narula, S.C. and C.A. Ho. (1980). “Degree Constrained Minimum Spanning Tree.” *Computers and Operations Research* 7, 239–249.
[2] Collins, N., R. Eglese, and B. Golden. (1998). “Simulated Annealing: An Annotated Bibliography.” *American Journal of Mathematical and Management Sciences* 8(3–4), 209–307.
[3] G. Zhou, M. Gen, "Approach to the degree-constrained minimum spanning tree problem using genetic algorithms", *Engineering Design and Automation*, vol. 3, no. 2, pp. 156-165, 1997.
[4] G. Zhou, M. Gen, "A note on genetic algorithm approach to the degree-constrained spanning tree problems", *Networks*, vol. 30, pp. 105-109, 1997.
[5] J. W. Moon. (1967). “Various proofs of cayley's formula for counting trees.” A Seminar on Graph Theory, pages 70.
[6] Palmer, C.E. and A. Kershenbaum. (1994). “Representing Trees in Genetic Algorithms.” Technical report, IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, 1994.

[7] Gen, M., G. Zhou, and M. Takayama. (1998). “A Comparative Study of Tree Encodings on Spanning Tree Problems.” IEEE Conference on Evolutionary Computation.

[8] T. C. Hu. (1974). “Optimum communication spanning trees.” SIAM Journal on Computing, 3(3), 188-195.

[9] Gabow, H.N. (1978). “Good Algorithm for Smallest Spanning Trees with a Degree Constraint,” Networks 8(3), 201–208.

[10] Boldon, B., N. Deo, and N. Kumar. (1996). “Minimum-Weight Degree-Constrained Spanning Tree Problem: Heuristics and Implementation on an Simd Parallel Machine.” Parallel Computing 22(3), 369–382.