IEEE Int. Workshop VLSI Design & Video Tech.
Suzhou, China, May 28~30, 2005

A New Genetic Algorithm for the Degree-Constrained Minimum Spanning Tree Problem

Li-Xia Han and Yuping Wang
Faculty of Science, Xidian University
Xi'an, 710071, China
Fu-Ying Guo
Computer Basic Teaching and Experiment Center
Xidian University, Xi'an, 710071, China

ABSTRACT

A novel genetic algorithm for the degree-constrained minimum spanning tree (d-MST) is proposed in this paper. In the proposed algorithm, a candidate solution is directly represented by set of the edges contained in the corresponding spanning tree, and a specific initialization scheme, a heuristic crossover and a mutation operator are designed, respectively. Moreover, both the crossover operator and the mutation operator are very easy to execute, and the offspring generated by them always represent feasible spanning trees. Furthermore, the global convergence of the proposed algorithm to globally optimal solution with probability one is proved. At last, the simulated results show the effectiveness of the proposed algorithm.

1. INTRODUCTION

The problem of identifying a minimum spanning tree of a connected, undirected graph is a classical combinatorial optimization problem of great importance in communication network design problem and other network-related problem. When the graph's costs are fixed, and the search is unconstrained, the well-known algorithm of Kruskal [1] or Dijkstra [2] algorithm can find MST effectively in polynomial time. In real networks, however, only spanning trees that satisfy some additional constraint are feasible. Most such problems are NP-hard, and we can apply exact optimization algorithms only to small instances of them. For large instances, we turn to heuristic techniques, including genetic algorithm (GAs) [3~6].

In the proposed algorithm, a candidate solution is directly represented by set of the edges contained in the corresponding spanning tree, and a special initialization scheme, a heuristic crossover and a mutation operator are then designed. Moreover, both the crossover operator and the mutation operator are very easy to execute, and the offspring generated by them always represent feasible spanning trees. Furthermore, the global convergence of the proposed algorithm to globally optimal solution with probability one is proved. At last, the simulated results show the effectiveness of the proposed algorithm. The d-MST problem can be stated as follows: Let $G=(V,E)$ be a complete graph of n nodes in which each edge (i,j) has a weight w_{ij}. The problem is to determine a spanning tree of the minimum total edge weight (sum of the weights of edges is minimum) such that at each node i its degree d_i cannot be greater than a given value d.

2. New Genetic Operators

In the proposed algorithm, the encoding scheme is directly representing a spanning tree by set T of edges forming the spanning tree. A genetic algorithm needs an initial population of diverse genotypes. For the edge set representation, we therefore need an algorithm that generated spanning trees. A technique based on Dijkstra's algorithm to generate random spanning tree is presented.

2.1. Creating random spanning trees

Dijkstra's algorithm [2] greedily builds a minimum spanning tree from a start node by repeatedly appending the lowest cost edge that joins a new node to the growing tree. Using it we can only obtain one spanning tree. However, if each new edge is chosen at random rather than according to its cost, then multiple spanning trees can be obtained. This produces a scheme we call Random Dijkstra (RD).

Unfortunately, RD yields trees of some structures with much higher probability than others. The number of distinct spanning trees of a complete graph on n nodes is n^{n-2}, and the probability to generate each of them by the uniform algorithm is

0-7803-9005-9/05/$20.00 ©2005 IEEE

$$p_{uniform} = \frac{1}{n^{n-2}}.$$

The probability that RD generates a spanning tree with degree $(n-1)$, denoted by p_{n-1}, is

$$p_{n-1} = \frac{2}{n!} > p_{uniform} \quad (n > 4).$$

While the probability that RD generates a Hamilton path, denoted by p_2, is

$$p_2 = \frac{2^{n-1}}{(n-1)!n!} < p_{uniform} \quad n \geq 4.$$

For the degree-constrained spanning tree, if each new edge is chosen at random without violating the degree constraint, a new initialization scheme called Degree-constrained Random Dijkstra (DRD) can be derived.

2.2 Crossover

Let $Nb(i,nb)$ be a set of the first nb shortest edges incident to node i, that is, $Nb(i,nb) = \{(i, j_{ik}) \mid k = 1 \sim nb\}$

Where j_{ik} is the k-th nearest node adjacent to node i. Let

$Nb(n) = \{Nb(i,nb), i = 1 \sim n\}$.

In this paper, we propose a novel crossover operator called KEX. KEX is modified from CX (also called UX) in [14], In the CX, the information contained in both parents is preserved. KEX also adopts this technique except for preserving only good genes from common edges of parents and constructing offspring based on the idea of Kruskal algorithm [1]. In the implementation, if the cost w_{ij} of edge (i,j) is the lowest one among all edges incident to node j or the lowest one among all edges incident to node i, it is called a "good" gene. The details of KEX can be described in the following steps.

Step 1. Find the common genes (edges) of both parents T_1 and T_2.

Step 2. Inherit "good" genes from $T_1 \cap T_2$ to form a graph T containing all good genes and all nodes.

Step 3. Complete a spanning tree with edges from $E_0 = Nb(n) \cup T_1 \cup T_2$ using the idea of random Kruskal algorithm as follows:

Choose an edge $e = (u,v) \in E_0$ at random and delete this edge from E_0. If both nodes u and v satisfy the degree constraint and $T \cup e$ contains on circle, then edge e is added to T. The process continues until T is a spanning tree.

This operator is simple and efficient; it favors cheap edges without excluding expensive ones, and it avoids complex repair and penalty mechanisms.

2.3. Mutation

The mutation operator favors the insertion of some low-cost edges and the deletion of other high-cost edges lying on the cycle introduced by the insertion if possible.

Choose a node i randomly whose degree is less than degree constraint, and identify a rank by the random variable $r = \lfloor |N(0, \beta n)| \rfloor \bmod n+1$,

Where $N(0, \beta n)$ is a normally distributed random variable with mean 0 and standard deviation βn. Then insert the edge (i,j) to the tree, where node j is the r-th nearest node to node i. Then there must be a circle in $T \cup \{(i,j)\}$. The deletion can be implemented in two cases.

a) If the degree of node j is less than degree constraint, then use roulette wheel selection (use the edge length to define the selection probability) to select the edge to be deleted in the circle;

b) Otherwise, then delete the edge e connected to node $j (e \neq (i,j))$ in the circle.

The process is repeated n times, and each node is selected as node i with the same probability. However, we may select a node j such that edge (i,j) is already in T, or there exists no such node j. In these cases, we throw away j and determine another j.

The heuristic mutation operator favors low-cost edges to be inserted with higher probability, while high-cost edge of the circle to be deleted with higher probability. The parameter β controls the strength of the schemes biased toward low-cost edge.

3. A New Genetic Algorithm for *d*-MST

Algorithm (NMST)

Step 1. (Initialization) Choose population size N, proper crossover probability p_c and mutation probability p_m, respectively. Use DRD to generate initial population $P(0)$. Let the generation number $t = 0$.

Step 2. (Crossover) Choose the parents for crossover from $P(t)$ with probability p_c. If the number of parents chosen is odd, then randomly choose additional

one from $P(t)$. Afterwards, randomly match every two parents as a pair and use the proposed crossover operator (KEX) to each pair to generate two offspring. All these offspring constitute a set denoted by O_1.

Step 3. (Mutation) Selection the parents for mutation from set O_1 with probability p_m. For each chosen parent, the proposed mutation operator is used to it to generate a new offspring. These new offspring constitute a set denoted by O_2.

Step 4. (Selection) Select the best N individuals among the set $P(t) \cup O_1 \cup O_2$ as the next generation population $P(t+1)$, let $t = t+1$.

Step 5. (Termination) If termination conditions hold, then stop, and keep the best solution obtained as the approximate global optimal solution of the problem; otherwise, go to step 2.

4. Global Convergence

To analyze the properties of genetic algorithms easily, we use a brief and general framework to describe genetic algorithm as follows: in each iteration the population is modified by a number of successive probabilistic transformation. Evidently, the resulting new population only depends on the state of the current population in a probabilistic manner. This property, known as the Markov property, reveals that Markov processes are appropriate models for the probabilistic behavior of genetic algorithms. Notice that the deterministic concept of "the convergence to the optimum" is not appropriate because the state transitions of a genetic algorithm are of stochastic nature. In order to clarify the exact semantic of a phrase like "the convergence to the global optimum" one has at first to define the exact stochastic convergence.

Let S is the search space of the degree-constrained MST problem. In order to prove the global convergence of the algorithm with probability one, it is required to introduce the following concept:

Definition 1. Let $T^* \in S$ denote the chromosome that corresponds to a minimum spanning tree. If $\Pr ob\{\lim_{t\to\infty} T^* \in P(t)\} = 1$, then the proposed genetic algorithm is called to converge to the global optimal solution with probability one, where $prob\{\}$ represents the probability of random event $\{\}$.

Definition 2. For two chromosomes a and b if $\Pr ob\{MC(a) = b\} > 0$, then chromosome b is called to be reachable from chromosome a by crossover and mutation, where $MC(a)$ represents the offspring that was generated from a by crossover operator and mutation operator.

Bäck in [7](pp. 129) has proved that if a genetic algorithm with a finite search space S satisfies the following conditions:

a) For any two chromosomes $a, b \in S$, b is reachable from a by crossover and mutation.
b) The population sequence
$P(0), P(1), \cdots, P(t), \cdots$
is monotone, i.e., for $\forall t$ we have
$$\min\{f(x) | x \in P(t+1)\} \le \min\{f(x) | x \in P(t)\}$$

It will converge to global optimal solution with probability one.

For the proposed algorithm NMST, we have the following conclusion:

Theorem 2 For any two chromosomes $a, b \in S$, b is reachable from a by crossover and mutation.

Theorem 3 The proposed genetic algorithm (NMST) converges to the global optimal solution with probability one.

Proof: For any two chromosomes $a, b \in S$, It can be seen from theorem 2 that b is reachable from a by crossover and mutation operator. It also can be seen from the selection scheme at step 4 of the proposed algorithm (NMST) that the population sequence
$P(0), P(1), \cdots, P(t), \cdots$
is monotone. Thus the conclusion is true.

5. Simulation Results

We execute the proposed algorithm (denoted by NMST) to solve seven benchmark problems. Due to the lack of standard benchmark problems at hand, we use the TSP problems: bayg29 (R1), dantzig42 (R2), eil51 (R3), pr76 (R4), eil101 (R5), gr202 (R6) and Lin318 (R7) as substitutes. We identified the minimum spanning tree employing the Dijkstra minimum spanning tree algorithm for each of these benchmark problems, and recorded its minimum total weight. Let $d_{\min}$ denote the degree of minimum spanning tree. If degree constraint is $d_i \le d$ satisfying $d > d_{\min}$, then the search space for d – MST contains the minimum spanning tree obtained by Dijkstra algorithm. Thus, to identify the efficiency of NMST, we only need to check whether NMST can find the solution as same as that obtained by Dijkstra 's algorithm with $d > d_{\min}$.

The simulations are carried out on an Intel Pentium

III 200-MHZ PC and we program NMST in Matlab language. For each benchmark problem, we perform 50 independent executions. We record the following data:

◆The weight of the minimum spanning tree obtained by Dijkstra algorithm, denoted by Opt.

◆The best, the worst, the average and the standard deviation of spanning tree weights over 50 runs, denoted by Best, Worst, Ave and σ, respectively.

◆The number of trials that found an optimal spanning tree, denoted by times.

Table 1 Results obtained by proposed algorithm: NMST

Problems	Opt	Best	Worst	Ave	σ	times
R1	7341	7341	7341	7341	0	50
R2	569	569	569	569	0	50
R3	375	375	375	375	0	50
R4	87217	87217	87217	87217	0	50
R5	551	551	551	551	0	50
R6	384	384	384	384	0	50
R7	37906	37906	37934	37908.8	0.2656	45

Table 1 summaries the results found by the proposed algorithm (NMST) and the times of runs among total 50 runs in which the proposed algorithm find these optimal solutions for problem R1~R7. It can be seen from the table 1 that for each of these problems the best solution found by NMST is the optimal solution. It can also be seen that the proposed algorithm finds the optimal solutions every run for all test problems except R7, For R7, although NMST finds the optimal solutions in 45 runs among the total 50 runs, the average lengths (Ave) for problem R7 is only a little bit worse than the optimal solution, moreover, the standard deviation is small. This illustrates that the solutions found in all runs for each of these problem are of high quality. Thus the proposed algorithm is effective and stable.

6. Conclusions

A novel genetic algorithm for the degree-constrained minimum spanning tree is proposed in this paper. In the proposed algorithm, a special initialization scheme, a heuristic crossover and a mutation operator are designed. Furthermore, the global convergence of the proposed algorithm to globally optimal solution with probability one is proved. At last, the simulated results show the effectiveness of the proposed algorithm.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 60374063).

7. REFERENCES

[1] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem", in Proc. Amer. Math. Soc.,vol. 7, pp. 48-50, 1956.

[2] E. W. Dijkstra, "A note on two problems in connection with graphs", *Numerische math.*, vol. 1, pp. 269-271, 1959.

[3] G. Zhou and M. Gen, "Genetic algorithm on leaf-constrained Minimum spanning tree problem", *Beijing Math.*, vol. 7, no. 2, pp. 50-62, 1998.

[4] H. Chou, G. Premkumar and C. H. Chu, "Genetic algorithms for communications network design-An empirical study of the factors that influence performance", *IEEE Transaction on Evolutionary Computation*, vol. 5, pp. 236-249, June 2001.

[5] J.Knowles and D.Corne, "A new evolutionary approach to The degree constrained minimum spanning tree problem", *IEEE Transaction on Evolutionary Computation*, vol.4, pp. 125-134, Apr. 2000.

[6] Günther R. Raidl and Bryant A. Julstrom, "Edge Sets: an Effective evolutionary coding of spanning trees", *IEEE Trans. Evolutionary Computation*, vol. 7, no. 3, pp. 225-239, 2003.

[7] Thomas Bŭck, "Evolutionary algorithms in theory and Practice", *New York*: Oxford University Press, 1998