

An Improved Genetic Algorithm for the Reliability Optimization of Computer Networks

Hai-Feng Wang

School of Information Science and Engineering
Jiaozuo Teachers College, Jiaozuo 454000, P.R. China

Abstract—The rapid development of computer network technology and the popularization of its application in various fields lead to the more and more complex networks. For this reason, an optimization approach and application is proposed to the reliability of computer networks using improved genetic algorithm. Experimental results suggest that this proposed approach is feasible, correct and valid.

Keywords-Reliability, Computer Network, Genetic Algorithm

I. INTRODUCTION

The rapid development of computer networking technology and the popularization of its application in various fields lead to continuous increase of network user nodes, increasingly larger network scale and more and more complex network. Meanwhile, people’s dependence on the computer network increases continuously and any failure of computer network system will cause huge loss and even immeasurable loss. Therefore, the network reliability becomes one of the important indicators that measure the comprehensive performance of computer network system. How to maximize the network reliability while reducing the total cost of network link also becomes a focus issue concerned by network designers and users. The traditional solving method is hard to solve the complex optimization NP problem. In view of this problem, in this paper, the genetic algorithm used for the computer network reliability optimization computing is put forward and the implementation of the algorithm is realized with examples.

II. PROBLEM FORMULATION

The assumptions of this problem can be summarized as follows.

- (1) The computer network is described by the mathematical figure $G=(N,L)$, and there shall be one direct link at most between any two nodes in figure G ;
- (2) The reliability of network link medium is unrelated to the medium length;
- (3) The network nodes are free from any failure;
- (4) There are only two working states of network and its links: normal and failure, and the occurrence of failure is independent mutually.

The cost matrix of network link medium is shown in equation (1), where C_0 is the cost matrix; and c_{ij} is the cost of link medium between node i and j ($1 \leq i \leq m, 1 \leq j \leq n$).

$$C_0 = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$$

The mathematical model of network link medium cost is shown as follows.

$$\begin{aligned} \text{Min } C &= \sum_{i=1}^N \sum_{j=1}^i c_{ij} g_{ij} \\ \text{s. t. } D_{iaij} &\leq \alpha \quad (i, j = 1, \dots, N) \\ \sum_{j=1, j \neq i}^N g_{ij} &\geq \beta \quad (i = 1, \dots, N) \end{aligned}$$

Where

$$g_{ij} = \begin{cases} 1 & \text{There is a direct link between node } i \\ & \text{and } j \\ 0 & \text{There is no direct link between node } i \\ & \text{and } j \end{cases}$$

C is the total network cost, N is the total network nodes, $D_{(a)}$ is the number of logical shortest link medium between node i and j , and α and β are the network node reliability constraint constants.

The reliability matrix of network link medium is shown as the following equation, where R_0 is the reliability matrix; and r_{ij} is the reliability of link medium between node i and j ($1 \leq i \leq m, 1 \leq j \leq n$).

$$R_0 = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}$$

In $G=(N,L)$, there may be only a subset L' of L is in normal state; when $L' \subseteq L$, the reliability of all nodes that are in normal state is $\prod_{i=1}^N P(n_i)$, where n_i mean node i , and $P(l_i)$ is the reliability of i link medium. The problem of computer network reliability optimization computing can be abstractly described as: maximize R , and make $C \leq C_0$. The mathematical model of computer network reliability is shown as follows:

$$R = \prod_{i \in L} P(l_i) \cdot \prod_{i \in (L \setminus L')} (1 - P(l_i)) \cdot \prod_{i=1}^N P(n_i)$$

Where Ω is the set of all network normal states and l_i is the i link medium.

III. THE PROPOSED METHOD

A. Population Initialization

(1) Divide the feasible region $[L, U]$ of the problem to be optimized into B subspaces according to the following equation

$$u_k - l_k = \max_{1 \leq i \leq n} \{u_i - l_i\}$$

$$\begin{cases} L_i = L + (i-1)[(u_k - l_k) / B]1_k \\ U_i = U + (B-i)[(u_k - l_k) / B]1_k \end{cases}, \quad i = 1, 2, \dots, B$$

Here, $L = [l_1, l_2, \dots, l_n]^T$ and $U = [u_1, u_2, \dots, u_n]^T$ respectively denote the lower boundaries and upper boundaries of n independent variables of the problem. B denotes the design parameter. 1_k is the n -dimensional vector of which the k^{th} bit is 1 and other bits are 0. L_i and U_i denote n -dimensional vectors respectively similar to L and U . In this way, the feasible region of the problem can be divided into B subspaces, namely $[L_1, U_1], [L_2, U_2], \dots, [L_B, U_B]$.

(2) Discretize each independent variable in each subspace according to the following equation. Suppose the domain of the independent variable x_i is $[l_i, u_i]$, then x_i can be quantized into Q_1 (the design parameter) levels $a_{i1}, a_{i2}, \dots, a_{iQ_1}$, and the detailed computation method for a_{ij} is

$$a_{ij} = \begin{cases} l_i & j = 1 \\ l_i + (j-i)[(u_i - l_i) / (Q_1 - 1)] & 2 \leq j < Q_1 \\ u_i & j = Q_1 \end{cases}$$

(3) Select M_1 chromosomes from each subspace. First construct the orthogonal table $L_{M_1}(Q_1^N) = [a_{ij}]_{M_1 \times N}$ in which N denotes the dimensionality of the problem, $M_1 = Q_1^{J_1}$, and J_1 denote the positive integers that satisfy the condition $\frac{Q_1^{J_1} - 1}{Q_1 - 1} \geq N$. Then select M_1 combinations from the Q_1^N ones to form M_1 chromosomes.

(4) Select the best G (the size of the initial population) chromosomes from the $M_1 B$ potential ones to form the initial population according to their fitness value.

B. Crossover Operation

Select two parent chromosomes for crossover operation according to the crossover probability. Suppose that the two selected parents are

$$\begin{cases} p_1 = (p_{11}, p_{12}, \dots, p_{1N}) \\ p_2 = (p_{21}, p_{22}, \dots, p_{2N}) \end{cases}$$

and the defined solution space $[l_{parent}, u_{parent}]$ is

$$\begin{cases} l_{parent} = [\min(p_{11}, p_{21}), \min(p_{12}, p_{22}), \dots, \min(p_{1N}, p_{2N})] \\ u_{parent} = [\max(p_{11}, p_{21}), \max(p_{12}, p_{22}), \dots, \max(p_{1N}, p_{2N})] \end{cases}$$

(2) Discretize the solution space $[l_{parent}, u_{parent}]$ of two parent individuals for crossover operation into Q_2 parts.

(3) Select some independent variables that will suffer crossover operations. The number of potential child individuals generated by each pair of parent individuals should be controlled to avoid a huge evaluation on the populations during the crossover operation. In this paper, the crossover only operates on F genes of the parent chromosome. Discretize these F independent variables in each subspace.

(4) Select potential child points from the solution space of parents according to the orthogonal table. First generate the orthogonal table $L_{M_2}(Q_2^F) = [b_{ij}]_{M_2 \times F}$ in which Q_2 is an odd number, $M_2 = Q_2^{J_2}$, and J_2 is the smallest positive integer that satisfies the condition $\frac{Q_2^{J_2} - 1}{Q_2 - 1} \geq F$. Then select

M_2 combinations from these Q_2^F ones to form M_2 potential child individuals.

(5) Select two with the best fitness value from the M_2 potential child individuals and two parents as the result of this crossover operation.

(6) If the number of the implemented crossover operations has reached the pre-set value, stop crossover immediately, or turn to (1).

C. Mutation Operation

(1) Randomly select a parent chromosome for mutation operation according to the mutation probability.

(2) Obtain the mutated child chromosomes according to the fine perturbation method. Here, the perturbation means generating four mutated child chromosomes through tuning the value of the selected parent gene respectively to its original $1-2\sigma$, $1-\sigma$, $1+\sigma$ times and $1+2\sigma$ times.

(3) Select the best one of the parent and child chromosomes as the result of this mutation operation.

(4) If the number of the implemented mutation operations has reached the preset value, stop mutation immediately, or turn to (1).

IV. EXPERIMENTAL RESULTS

Below is an example of computer network reliability optimization computing. With its mathematical model as mentioned above, the cost matrix C_0 and reliability matrix R_0 of network link medium are shown as follows.

$$C_0 = \begin{bmatrix} 0 & 5 & 8 & 12 & 14 & 7 \\ 5 & 0 & 9 & 4 & 8 & 11 \\ 8 & 9 & 0 & 5 & 10 & 9 \\ 12 & 4 & 5 & 0 & 3 & 7 \\ 14 & 8 & 10 & 3 & 0 & 6 \\ 7 & 11 & 9 & 7 & 6 & 0 \end{bmatrix}$$

$$R_0 = \begin{bmatrix} 0 & 0.98 & 0.95 & 0.99 & 0.999 & 0.96 \\ 0.98 & 0 & 0.97 & 0.988 & 0.996 & 0.99 \\ 0.95 & 0.97 & 0 & 0.96 & 0.95 & 0.998 \\ 0.99 & 0.988 & 0.96 & 0 & 0.93 & 0.92 \\ 0.999 & 0.996 & 0.95 & 0.93 & 0 & 0.97 \\ 0.96 & 0.99 & 0.998 & 0.92 & 0.97 & 0 \end{bmatrix}$$

The network nodes N are 6, the network node reliability constraint constant α and β are 2 respectively, and the iteration number of genetic manipulation is 100. By simulation solving, the total cost of network link medium is 45, and the maximum network reliability is 0.885.

V. CONCLUSIONS

In this paper, an improved genetic algorithm is used in the computer network reliability optimization computing to achieve the goal of reducing the cost of network link medium and improving the network reliability, with simple algorithm, easy programming and good readability. The simulation example proves that the genetic algorithm is of practical value in solving the said problem above.

REFERENCES

- [1] Y.N. Li, L.Z. Zhang, J. Shu, et al., Application of Expert Knowledge Adopted Genetic Algorithm to Optimization of Reactive Power Planning, *Power System Technology*, 2001, 25(7), pp. 14-17
- [2] F. Liu, G.Z. Zeng. Study of Genetic Algorithm with Reinforcement Learning to Solve the TSP, *Expert Systems with Applications*, 2009, 36(2), pp. 6995-7001
- [3] S. Louis, G. Li, Augmenting Genetic Algorithms with Memory to Solve Traveling Salesman Problems, in the Proceedings of the Joint Conference on Information Sciences, Durham, North Carolina: Duke University Press, 1997, pp. 108-111
- [4] S.J. Louis, G. Li, Case Injected Genetic Algorithms for Traveling Salesman Problems, *Information Science*, 2000, 122(2-4), pp. 201-225
- [5] S.J. Louis, J. McDonnell, Learning with Case-Injected Genetic Algorithms, *IEEE Transactions on Evolutionary Computation*, 2004, 8(4), pp. 316-328
- [6] R.S. Michalski, Learnable Evolution Model: Evolution Process Guided by Machine Learning, *Machine Learning*, 2000, 38(1), pp. 9-40
- [7] R.S. Michalski, J. Wojtusiak, K.A. Kaufman, Intelligent Optimization via Learnable Evolution Model, in the Proceedings of the Eighteenth IEEE International Conference on Tools with Artificial Intelligence, Arlington, VA: IEEE Press, 2006, pp. 332-335
- [8] R.S. Michalski, J. Wojtusiak, K.A. Kaufman, Progress Report on Learnable Evolution Model, Fairfax, VA: George Mason University, 2007
- [9] K. Rasheed, H. Hirsh, Using Case-Based Learning to Improve Genetic-Algorithm-Based Design Optimization, in the Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI: IEEE Press, 1997, pp. 513-520
- [10] R.G. Reynolds, an Introduction to Cultural Algorithms, in the Proceedings of the Third Annual Conference on Evolutionary Programming, River Edge, NJ: World Scientific, Singapore, 1994, pp. 131-139