Research on a Novel Multicast Routing Hybrid Optimization Algorithm and Its Application

Dejia Shi1, Weijin Jiang1,2,3, Danping Xu3
1School of Computer and Electronic Engineering, Hunan University of Commerce, Changsha, P. R. China
2School of Computer, Wuhan University of Technology, Wuhan, P. R. China
3The College of Information Engineering, Xiangtan University, Xiangtan, P. R. China
nudbyyl@126.com

Abstract

The problem of multicast routing with multiple QoS constraints is NP complete problem. Most of the multimedia applications require strict QoS guarantee during the communication between a single source and multiple destinations. This gives rise to the need for an efficient QoS multicast routing strategy. Determination of such QoS-based optimal multicast routes basically leads to a multi-objective optimization problem, which is computationally intractable in polynomial time due to the uncertainty of resources in high-performance networks. This paper describes a network model for researching the routing problem and we propose a new multicast tree selection algorithm based on genetic algorithms to simultaneously optimize multiple QoS parameters. Based on QoS constrains such as delay, delay jitter, bandwidth and packet loss metrics this paper describes a network model suitable for investigating the routing problem and presents a multicast routing algorithm with multiple QoS constraints based on GA and TS hybrid strategy. This algorithm takes advantage of GA and TS and overcomes the shortcomings of GA used in solving the multicast routing problem with multiple QoS constraints—poor climbing ability and immature convergence. The simulation results show that the proposed algorithm is able to find a better solution, fast convergence speed and high reliability. It can meet the real-time requirement in multimedia communication networks. The scalability and the performance of the algorithm with increasing number of network nodes are also quite encouraging.

1. Introduction

Multicast services have been used by a variety of continuous media applications. For example, the multicast backbone of the Internet has been used to transport real time audio/video for news, entertainment, video conferencing, and distance learning[1-3]. The provision of Quality-of-Service (QoS) guarantees is of utmost importance for the development of the multicast services[4-7]. Multicast routing has continued to be a very important research issue in the areas of networks and distributed systems. It attracts the interests of many people. QoS multicast routing relies on state parameters specifying resource availability at network nodes or links, and uses them to find paths with enough free resources[6,8].

The traditional multicast routing protocols, e.g., CBT and PIM[5-7], were designed for best-effort data traffic. They construct multicast trees primarily based on connectivity. Such trees may be unsatisfactory when QoS is considered due to the lack of resources. Several QoS multicast routing algorithms have been proposed recently. Some algorithms[1,2,5,6] provide heuristic solutions to the NP-complete constrained Steiner tree problem, which is to find the delay-constrained least-cost multicast trees. These algorithms however are not practical in the Internet environment because they have excessive computation overhead, require knowledge about the global network state, and do not handle dynamic group membership. In addition, an extra control element, called manager router, is introduced to handle the join requests of new members. Multicast routing and its QoS-driven extension are indispensable components in a QoS-centric network architecture[1,5,6,9]. Its main objective is to construct a multicast tree that optimizes a certain objective function (e.g., making effective use of network resources) with respect to performance related constraints.

In recent years some researches have started using evolutionary algorithms to find near-optimal solutions for different networking problems, like QoS-Routing[6,7,9]. More recently, researches in determining QoS-based multicast routes clearly demonstrate the power
of genetic algorithms to get a near-optimal solution satisfying the QoS requirements in computationally feasible time\cite{3,4,8,10}. A little careful insight into these above optimization schemes reveals that all of them suffer from the same drawback: multiple objectives are combined to form a scalar single-objective function, usually through a linear combination of multiple attributes. In these cases the solution not only becomes highly sensitive to the weight vector but also demands the user to have certain knowledge (e.g. priority of a particular objective, influence of a parameter over another etc.) about the problem. Moreover, in case of multi-objective optimization, a unique solution that optimizes all the objectives simultaneously will rarely, if at all, exist in practice. However, the genetic algorithm (GA) can be readily modified to deal with multiple objectives by incorporating the concept of Pareto domination in its selection operation\cite{11,12}.

TS (Tabu Search) algorithm is a kind of algorithm to do overall domain search. It makes use of partial domain search mechanism and corresponding tabu rules to avoid circuit search. Breaking the tabu level to releases some excellent states that are tabooed it assures the diverse and effective search and overcomes the shortcoming of premature to realize overall optimization. But TS depends on initial solution deeply.

In this paper we will concentrate on determining multicast routes from a source to a set of destinations with strict end-to-end delay requirements and minimum bandwidth available. The goal of this paper is to develop an algorithm to find out QoS-based multicast routes by simultaneously optimizing end-to-end delay, bandwidth provisioning for guaranteed QoS and proper bandwidth utilization without combining them into a single scalar optimization function. This paper put forwards a QoS multicast routing algorithm based on GA and TS hybrid strategy aiming to the merits and shortcomings of QoS multicast routing. This algorithm introduces TS memory function to GA search and establishes new mutation operator and regrouping operator to overcome the shortcoming of multicast routing protocol by the use of GA premature convergence.

2. Network Model

A network is usually represented as a weighted digraph $G = (N, E)$, where $N$ denotes the set of nodes and $E$ denotes the set of communication links connecting the nodes. $|N|$ and $|E|$ denote the number of nodes and links in the network respectively. Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes\cite{9-10}. We consider the multicast routing problem with bandwidth and delay constraints from one source node to multi-destination nodes. Let $M = \{n_0, u_1, u_2, \ldots, u_m\} \subseteq N$ be a set of from source to destination nodes of the multicast tree. Where $n_0$ is source node, $U = \{u_1, u_2, \ldots, u_m\}$ be a set of destination nodes. Multicast tree $T = (N_T, E_T)$, where $N_T \subseteq N$, $E_T \subseteq E$, there exists the path $P(n_0, d)$ from source node $n_0$ to each destination node $d \in U$ in $T$.

A network can be modeled as a weighed graph $G = (V, E)$, where $V$ and $E$ are the sets of nodes and links, $s \in V$ is multicast source node, $M \subseteq \{V - \{s\}\}$ is multicast destination nodes sets, $R_s$ represents positive real number aggregate, $R^0$ represents non-minus real number aggregate. As to any link $e \in E$ four measurement are defined: delay function “$\text{delay}(e): E \rightarrow R^0$ ”, cost function “$\text{cost}(e): E \rightarrow R^0$ ”, bandwidth function “$\text{bandwidth}(e): E \rightarrow R^0$ ”, delay jitter function “$\text{delay_jitter}(e): E \rightarrow R^0$ ”. As to any network node $n \in V$, four measurement are also defined: delay function “$\text{delay}(n): V \rightarrow R^0$ ”, cost function “$\text{cost}(n): V \rightarrow R^0$ ”, delay jitter function “$\text{delay_jitter}(n): V \rightarrow R^0$ ”, packet loss function “$\text{packet_loss}(n): V \rightarrow R^0$ ”. Then there exists following relation among source node “$s \in V$”, destination nodes sets “$M$”, multicast tree “$T(s, M)$” composed of “$s$” and “$M$”,

\begin{align}
\text{delay}(P_1(s, t)) &= \sum_{e \in P_1(s, t)} \text{delay}(e) + \sum_{n \in P_1(s, t)} \text{delay}(n) \quad (1) \\
\text{cost}(T(s, M)) &= \sum_{e \in T(s, M)} \text{cost}(e) + \sum_{n \in T(s, M)} \text{cost}(n) \quad (2) \\
\text{bandwidth}(P_1(s, t)) &= \min \{\text{bandwidth}(e), e \in P_1(s, t)\} \quad (3) \\
\text{delay_jitter}(P_1(s, t)) &= \sum_{e \in P_1(s, t)} \text{delay_jitter}(e) + \sum_{n \in P_1(s, t)} \text{delay_jitter}(n) \quad (4) \\
\text{packet_loss}(P_1(s, t)) &= 1 - \prod_{e \in P_1(s, t)} (1 - \text{packet_loss}(e)) \quad (5)
\end{align}

$P_1(s, t)$ is the routing path between source node “$s$” and destination node “$t$” of multicast tree $T(s, M)$. So QoS multicast routing problem can be described like this: in the network $G = (V, E)$, $s \in V$ is multicast source node, $M \subseteq \{V - \{s\}\}$ is multicast destination node set, delay function $\text{delay}(\ast) \in R_+$, cost function $\text{cost}(\ast) \in R_+$, bandwidth function $\text{bandwidth}(\ast) \in R_+$ and packet loss function $\text{packet_loss}(\ast) \in R_+$, seeking a multicast tree $T(s, M)$ which suffice following conditions:

\begin{enumerate}
\item delay constraint: $\text{delay}(P_1(s, t)) \leq D_{\text{max}}$
\item bandwidth constraint: $\text{bandwidth}(P_1(s, t)) \geq B_{\text{min}}$
\end{enumerate}
delay jitter constraint: \( \text{delay}_\text{jitter}(P_t(s, t)) \leq J_{\text{max}} \)

packet loss constraint: \( \text{packet}_\text{loss}(P_t(s, t)) \leq L_{\text{max}} \)

cost constraint: in the multicast tree sufficing above four conditions, \( \text{cost}(T(s, M)) \) is minimum.

\( P_t(s, T) \) is the routing path between source node “\( s \)” and destination node “\( t \)” of \( T(s, M) \). \( B_{\text{min}} \) is the bandwidth constraint, \( D_{\text{max}}, J_{\text{max}} \) and \( L_{\text{max}} \) are respectively delay constraint, delay jitter constraint and packet loss constraint.

3. Multicast routing algorithm based on hybrid evolution strategy

Before the production of initial population network \( G = (V, E) \) will be pretreated: First of all, the leaf nodes that are non-destination nodes with 1 degree will be removed. Then links that fails to suffice the bandwidth constraint will be eliminated. At this time, if multicast source node and destination node are not in the same connected graph, which means that the network cannot suffice bandwidth constraint, then the constraint should be relaxed and the processing should be done again. Finally a connected graph \( G' \) containing source node and destination node set is selected to be the network processed by the algorithm. After pretreatment bandwidth constraint conditions are eliminated and search space of the algorithm is reduced.

Production of initial population adopts random DFS(Depth-First Search). First of all, we start from source node and randomly select a corresponding node, then we connect these two nodes, and then select another correlative node (Attention: Avoid the appearance of loop. If loop comes into being, then select again.) until find all destination nodes. But not every individual in the initial population suffices QoS constraint requirements.

Since the underlying approach is based on multi-objective genetic algorithms (MOGA), our next step is to map the problem in a search space suitable to MOGA. Each of all the generated multicast trees is mapped to a string consisting of the sequence of nodes along the path from the source \( v_s \) to each of the destinations \( v_{d1}, v_{d2}, ..., v_{dk} \). To mark the end of a path from a source to a single destination, we use -1 as sentinel. Fig. 1 below gives a clear view of this scenario where a multicast tree is represented by a string. The set of all such strings constitute the initial population. The size of this population \( \text{popsize} \) depends on how the strings are created, which in turn depends on the network topology and the number of multicast destination nodes.

![Fig. 1 Representation of the multicast tree and its encoding scheme](image)

The individual with good performance has high fitness level, and the individual with bad performance has low fitness level. Let links be service queues where packets to be transmitted get serviced. For most cases this service can be assumed to follow Poisson distribution. The service time should follow an exponential distribution. Let the delay for link \( l \) be denoted by the variable \( d_l \) which is a random variable following exponential distribution with parameter equal to \( \lambda \). So the delay over a path consisting of \( k \) links would be the sum of \( k \) independent random variables all having the same exponential distribution and so would follow an Erlang-K distribution. From the definition of Erlang-K distribution we get that the probability that the delay \( (d_p) \) over a path \( P \) of length \( k \) is less than \( t \) is given by the following equation:

\[
\Pr(d_p < t) = \frac{\lambda^t e^{-\lambda t}}{(k-1)!}.
\]

From the classical probability theory we can say that the probability that the delay \( (d) \) of the selected multicast tree \( (T) \) will meet the specific delay constraint can be obtained by taking the product of delay over individual paths in that multicast tree:

\[
\Pr(d_T < t) = \prod_{p \in T} \Pr(d_p < t).
\]

To find an optimal path, our objective is to maximize this probability of satisfying delay requirements. The measure of the bandwidth guarantee can be obtained by assuming a similar model for the network links. If the service rate or the transmission rate, which is basically a measure of link bandwidth, is assumed to follow a poisson distribution, the probability that a link \( l \in E \) can provide a bandwidth of \( B \) is given by

\[
\Pr_l(B) = \frac{\lambda^l e^{-\lambda}}{B!}.
\]

We can now say that the probability with which the bandwidth guarantee of \( B \) is satisfied for an entire multicast tree \( (T) \) is given by:

\[
\Pr_T(B) = \prod_{l \in T} \Pr_l(B).
\]

The total residual bandwidth in the network after allocating bandwidth for a multicast \( T(s, M) \), is given by \( \Sigma_{l \in E} (c_l - b_l) \), where \( c_l \) is the capacity of a link \( l \in E \) and \( b_l \) is the bandwidth allocated for all the paths.
in the multicast \(T(s, M)\), along the link \(l\). Obviously, \(b_l\) is 0 if \(l \not\in p\) where \(p \in T\). The fraction of total bandwidth available as residual bandwidth is given as

\[
R(T) = \frac{\sum_{l \in M}(c_l - b_l)}{\sum_{l \in M}c_l}
\]  

(10)

The fitness sharing function of QMRGA can be defined as follows:

\[
f'(x_i) = \frac{f(x_i)}{m_i}
\]  

(11)

To incorporate this idea of fitness sharing we compute the value of niche count for every individual string present in the population, as:

\[
m_i = \sum_{j=1}^{\text{popsize}} SH[d_{s1}, d_{s2}]
\]  

(12)

where \(d_{s1, s2}\) is the distance between individuals \(s1\) and \(s2\) and \(SH[d_{s1, s2}]\) is sharing function. For simplicity, triangular sharing function has been used:

\[
SH[d_{s1, s2}] = \begin{cases} 1 - \frac{d_{s1, s2}}{\sigma_{\text{share}}} & d \leq \sigma_{\text{share}} \\ 0 & d > \sigma_{\text{share}} \\ \end{cases}
\]  

(13)

where \(\sigma_{\text{share}}\) is the niche radius, and it is a good estimate of minimal separation expected between the goal of solutions.

The phenotypic distance between two strings is nothing but the Euclidean distance between their different fitness values:

\[
d_{s1, s2} = \sqrt{(\sigma_{\text{delay}, s1, s2})^2 + (\sigma_{\text{bw}, s1, s2})^2 + (\sigma_{\text{bit}, s1, s2})^2}
\]  

(14)

where \(\sigma_{\text{delay}, s1, s2} = Pr(d_{s1, s2} < 0)\), \(\sigma_{\text{bw}, s1, s2} = Pr_{s1}(B) - Pr_{s2}(B)\), \(\sigma_{\text{bit}, s1, s2} = R(s1) - R(s2)\).

Similarly, we compute the niche radius \(\sigma_{\text{share}}\), some fraction of the maximum separation possible in the population, i.e.

\[
\sigma_{\text{share}} = \frac{\sqrt{(\sigma_{\text{delay}, \text{max}})^2 + (\sigma_{\text{bw}, \text{max}})^2 + (\sigma_{\text{bit}, \text{max}})^2}}{4}
\]  

(15)

where \(\sigma_{\text{delay}, \text{max}} = Pr_{\text{max}}(d < 0) - Pr_{\text{max}}(d < 0)\), \(\sigma_{\text{bw}, \text{max}} = Pr_{\text{max}}(B) - Pr_{\text{max}}(B)\), \(\sigma_{\text{bit}, \text{max}} = \max R - \min R\).

Fitness function is the criteria used by GA to judge whether the individual is good or not. The fitness function is:

\[
f(x) = \frac{\omega_1 \cdot f(d) + \omega_2 \cdot f(j) + \omega_3 \cdot f(l)}{\cos(T(s, M))}
\]  

(16)

where \(\omega_1, \omega_2, \omega_3, \omega_4\) are respectively weighted coefficient about cost, delay, delay jitter and packet loss. \(f(x), f(j), f(p)\) are like following:

\[
f(d) = \prod_{i \in M} F_i(\text{delay}(P_i(s, t)) - D_{\text{max}})
\]  

(17)

\[
F_i(\text{delay}(P_i(s, t)) - D_{\text{max}}) = \begin{cases} 1, & \text{delay}(P_i(s, t)) < D_{\text{max}} \\ \alpha, & \text{delay}(P_i(s, t)) \geq D_{\text{max}} \\ \end{cases}
\]  

(18)

\[
f(j) = \prod_{i \in M} F_i(\text{delay jitter}(P_i(s, t)) - J_{\text{max}})
\]  

(19)

\[
F_i(\text{delay jitter}(P_i(s, t)) - J_{\text{max}}) = \begin{cases} 1, & \text{delay jitter}(P_i(s, t)) < J_{\text{max}} \\ \beta, & \text{delay jitter}(P_i(s, t)) \geq J_{\text{max}} \\ \end{cases}
\]  

(20)

\[
f(p) = \prod_{i \in M} F_p(\text{packet loss}(P_i(s, t)) - L_{\text{max}})
\]  

(21)

\[
F_p(\text{packet loss}(P_i(s, t)) - L_{\text{max}}) = \begin{cases} 1, & \text{packet loss}(P_i(s, t)) < L_{\text{max}} \\ \sigma, & \text{packet loss}(P_i(s, t)) \geq L_{\text{max}} \\ \end{cases}
\]  

(22)

\[
F_i(x), F_j(x) \text{ and } F_p(x) \text{ are respectively penalty function about delay, delay jitter and packet loss. } \alpha, \beta \text{ and } \sigma \text{ are positive numbers less than } 1.
\]

This algorithm adopts roulette wheel selection. The selection probability of individual \(i\) with \(f_i\) fitness value is

\[
P_{\text{select}} = f_i / \sum_{j=1}^{N_{\text{pop}}} f_j, \quad P_{\text{select}} \text{ reflects the proportion between the fitness value of individual } i \text{ and the sum of all individual fitness value in the population.}
\]

Both the crossover and mutation operations can only be performed at the end of an existing path, i.e. immediately after a particular sentinel, represented by -1.

**Definition 1:** If \(A, B\) are a pair of neighbor nodes, and its corresponding side is \(e\); \(G1\) and \(G2\) are two disjunct subtree belonging to \(G'\); If \(A \in G1 \land B \in G2\), then correlation degree is defined like this:

\[
\text{Corre} = \frac{\alpha_1 \cdot \text{delay}(e) + \omega_2 \cdot \text{delay jitter}(e))}{\cos(T)}
\]  

(23)

According to the moves mentioned in 3.1, we firstly select paternal generation individual on the basis of random number and regrouping probability, then we make use of the chiasm of those paternal generation individual to produce filial generation. The chiasm is carried out like this: filial generation inherits the mutual side of two paternal generation individuals. So the filial generation may be no-connected composed of multi-subtree. Then each two no-connected subtree will be connected by node with high correlation degree to become a subtree. And this subtree is the filial generation chromosome.

Using TSR regrouping operator to regroup those filial generation individual produced previously by chiasm. TSR adopts taboo table \(TS(T)\) which is \(T\) in width. In this table chromosome’s fitness value are recorded, and its breaking taboo level is the average of paternal generation population’s fitness value. Supposed that the fitness value of filial generation individual \(i\) is \(f_i\) and its paternal generation population
average fitness is \( f_{\text{aver}} \), then TSR processes like following:

\[
\text{if } f > f_{\text{aver}}, \\
\text{then accept filial generation } i \\
\text{else if } i \notin \text{TS}(T) \text{ then accept filial generation } i \\
\text{else select the best paternal generation individual to enter next generation}
\]

modify TS(T)

By using taboo table TS(T), TSR can reduce the appearance of chromosome with low fitness value and keep the diversity of chromosome structure. So the premature of this algorithm can be avoided.

Both the crossover and mutation operations can only be performed at the end of an existing path, i.e. immediately after a particular sentinel, represented by -1.

TSM carries out mutation. Unlike normal mutation operator, TSM is a TS process which verifies motion value by evaluation function and decides to accept which motion output by motion value and taboo table.

When No. \( i \) chromosome is in the process of mutation, chromosome \( i \) is input, and an output is produced after TSM processing. TSM processing goes like this:

\[
\text{establish taboo table } T \text{ and initial solution } x_0 = x_i, \\
r = 0; \\
\text{while the ending conditions are not sufficed do} \\
\quad r = r + 1; \\
\quad \text{move } x_i; \\
\quad \text{update } x_i, \text{update } x_0, \text{update } T; \\
\quad \text{output } x_i
\]

Both the crossover and mutation operations can only be performed at the end of an existing path, i.e. immediately after a particular sentinel, represented by -1. Fig. 2 shows an example of the crossover operation. In case of mutation we just replace the part of the chromosome after the mutation point by a corresponding part of any other valid chromosome. To combine the good strings and simultaneously preserve the effective ones we have taken the probability of cross over as 0.2 and that of mutation as 0.02. Fig. 3 shows the overall operation of the mutation operation.

**Fig. 2** Overall operation of the crossover

**Fig. 3** Overall operation of the mutation

### 4. Simulations Results

The algorithm put forward in this paper is a kind of GA with hybrid strategy. First of all, it makes use of TSR and TSM produced by TS, and its processing is in fact the performing of TS Algorithm. They suffice the requirements of theorem 7.1\(^8\) and can get the optimal solution. Meanwhile the algorithm put forward in this paper suffices three conditions of astringency put forward by theorem 2.7\(^9\), so this algorithm can get the optimal solution.

C Language is used to carry out the algorithm in this paper and a simulation test about the network model in fig1 is carried out. In this test supposed that all multicast destination nodes have the same QoS constraints, and we only consider the characteristic of side but ignore the characteristic of node, and the model undergoes pretreatment of bandwidth constrain. The characteristic of each side in the fig is described by Quaternion \((D,J,B,C)\) which represents delay, delay jitter, bandwidth and cost.

Parameters in this test are defined in this way:

\[
N_{\text{gen}} = 30, \quad N_{\text{pop}} = 15, \quad P_c = 0.5, \quad P_m = 0.05, \quad \omega_f = 1, \omega_f = 0.5, \omega_f = 0.5, \omega_r = 0.3, \quad \omega_a = 0.5, \quad \beta = 0.5, \quad \sigma = 0.5.
\]

Fig.4 describes the movement curve of cost, delay, delay jitter with the operation of the algorithm when QoS constraint value \( D_{\text{max}} = 20, \quad J_{\text{max}} = 30, \quad B_{\text{min}} = 40. \) Fig.5 describes the movement curve of cost, delay, delay jitter with the operation of the algorithm when QoS constraint value \( D_{\text{max}} = 25, \quad J_{\text{max}} = 35, \quad B_{\text{min}} = 40. \) From Fig.4 and Fig.5 we can find out that this algorithm can get the optimal solution.

The initial status of link route in the network: the delay of link route is randomly distributed among \([5\text{ms}, 30\text{ms}]\), the bandwidth among \([60\text{Mbps}, 100\text{Mbps}]\), the price of the link route is randomly distributed among \([5, 30]\). Simulation experiments are performed over a network of 100 nodes, the source node and destination node of the multicast is randomly
chosen, the multicast destination node is less than the 20% of total nodes, delay constraint \(D\) is 60, Bandwidth constraint \(B=80\). The multicast QoS routing protocol, designed by us tries to maximize the probabilities of meeting end-to-end delay, bandwidth requirement and bandwidth utilization within a few generations by building the Pareto optimal fronts.

Fig. 5 Movement curve of cost, delay, delay jitter with the operation of the algorithm (\(D_{\text{max}}=25, J_{\text{max}}=35, B_{\text{min}}=40\)).

The multicast QoS routing protocol, designed by us tries to maximize the probabilities of meeting end-to-end delay, bandwidth requirement and bandwidth utilization within a few generations by building the Pareto optimal fronts.

5. Conclusions

This paper proposes a QoS multicast routing model and optimization algorithm based on bandwidth and delay constraints and gives the heuristic genetic algorithm of minimum-costs QoS multicast tree and bandwidth-delay constraint. Experiment represents that its convergent speed is fast and reliability. Especially in large network, the algorithm can greatly decrease routing computation time, satisfying the topology structure of real-time communication environment, high dynamic and the requirement of network structure of QoS multicast routing. QoS multicast routing is the foreland research project in networks and information technology. While interests of many people have got better results in single constraint (specially delay constraint) multicast routing, the results based on multi-constraint QoS multicast routing are not so satisfying. This algorithm can expand to multi-constraint’s QoS multicast routing problem based on this algorithm; the chromosome of fitness function alone can be changed and the delay constraint can be improved so as to make the algorithm widely applied.

Fig. 6 denotes the convergence comparison of cost with operation algebra of multicast tree generated by the two algorithms, this algorithm can speedily generate the optimal solution, furthermore, its advantage is more obvious when network scale is bigger, and bandwidth constraint is amplified. Repeat the simulation with increasing number of network nodes and observe the efficiency of our algorithm. As the network becomes highly condensed, our algorithm exhibits a more linear and stable pattern than existing scalar optimization algorithm. This approximate linearity of the curve in Fig. 7 corroborates the scalability of the algorithm.

Acknowledgement

This paper is supported by the Society Science Foundation of Hunan Province of China No. 07YBB239, and the Natural Science Foundation of Hunan Province of China No. 06JJ2033.

References


![Fig. 6 The effect of the cost with genetic generation](image)

![Fig. 7 Performance of the algorithm with increasing number of nodes](image)