

# 用一种改进遗传算法求解带时延约束的组播路由问题

蔡启博, 方建晓

(温州大学物理与电子信息学院, 浙江温州 325035)

**摘要:** 提出一种改进的遗传算法, 用以求解带时延约束的组播路由问题. 该算法综合考虑了路由费用和路由时延, 并给出了一种适应度评价函数, 在算法中采用了最佳个体保留、自适应交叉以及非线性排序选择等多种优化机制. 仿真结果表明, 该算法是可行的, 有效的.

**关键词:** 组播路由; 遗传算法; 时延约束

**中图分类号:** TN911    **文献标识码:** A    **文章编号:** 1006-0375(2007)05-0041-06

随着计算机网络技术的飞速发展, 诸如视频点播、电视会议、计算机协同计算等新的应用开始流行并将成为今后的发展趋势. 分析这类业务, 不难发现它们的共同点是通信数据量大, 时延要求尽可能小, 并有多点参与通信. 数据量大意味着需要更大的网络带宽, 时延约束则要求信息被更快地传送到接受方, 而多点意味着信息将被传输到多个目标. 点到点的通信方式要求在信源和信宿之间建立单独的路由, 如果在多点进行通信时采用这种方法, 信息将会在某些路径被重复传递, 造成网络带宽的浪费. 当通信量集中时甚至会造成某些通信链路的拥塞. 组播路由最初研究的是建立无约束的路由, 使得网络代价最小化. 其中基于构造 Steiner 共享树的路由算法以其实效性成为研究关注的中心.

由于构造一棵带时延约束的 Steiner 树已被证明是 NP 完全问题, 所以常规的启发式算法显得力不从心. Kompella 等人首先提出了求解代价最优和带时延约束的组播路由的启发式算法<sup>[1]</sup>, 在本文中称为 KPP1 和 KPP2 算法. Zhu.Q 等提出了一种受限的最短路径组播路由算法<sup>[2]</sup>, 该算法可以获得很低的路由费用, 但是计算复杂度很高, 且计算时间过长. Sun 和 Langendoerfer 在 Dijkstra 算法的基础上又提出了一种带时延约束的启发式算法<sup>[3]</sup>. 在文献[4]中 Slama 等对已有的一些算法进行了比较. 遗传算法具有并行搜索、群体寻优的特点, 已广泛用于解决具有 NP 难度的问题. 本文对遗传算法进行了改进, 提出一种适应度评价方法, 并结合多种优化机制. 仿真结果表明, 算法具有较好的计算结果和较快的收敛速度.

## 1 网络模型及问题的描述

通信网络可以被抽象为一个无向赋权图  $G = (V, E)$ , 其中  $V$  为网络中的节点,  $E$  代表节点之间的链路的集合. 两个非负的实值函数  $D(e)$  和  $C(e)$  分别表示链路时延和链路费用. 链路时延  $D(e)$  是

收稿日期: 2006-12-26

作者简介: 蔡启博(1979-), 男, 浙江温州人, 助理实验师, 研究方向: 通信工程

分组数据包通过链路的时间,暂不考虑分组在路由器中的缓冲排队等待路由的延时,链路费用函数  $C(e)$  则代表相应链路的资源的使用情况。

组播路由问题可以按如下方法进行描述:给定一个节点  $s(s \in V)$ , 一组目标节点  $d(d \subseteq V - s)$ , 组播路由的目标就是构建这样一棵以源节点  $s$  为树根, 遍历目标节点集合  $d$  中所有节点的组播树  $T$ , 使得信息从源节点沿着树枝被传送到所有的叶子节点(目标节点)。组播树的代价如下式所示:

$$C_T = \sum_{e \in T} C(e) \quad (1)$$

上式中的  $C_T$  表示组播树的代价,它等于组播树中所有树枝(也就是所有链路)上的费用的总和。目标节点集合  $d$  中的节点通常称之为组播组成员。网络中的所有不包含在  $\{s\} \cup d$  中的节点被称为 Steiner 节点。带时延约束的最小费用组播问题可以按以下方式进行描述:对于一个网络拓扑  $G = (V, E)$ , 一个源节点  $s(s \in V)$ , 一组目标节点  $d(d \subseteq V - s)$ , 以及时延的最大边界值  $\Delta$ , 我们的目标就是要找到一棵组播树  $T(T \in G)$ , 使得树的费用  $C_T$  最小, 而且

$$\sum_{e \in P_T(s, v)} D(e) < \Delta \quad \forall v \in d \quad (2)$$

其中  $P_T(s, v)$  代表从源节点到目标节点的路径, 上式表明要求所有的路由都要满足时延的约束, 换句话说, 网络的路由时延是指所有时延中的最大值。很显然, 这是一个带约束条件的优化问题, 属于 NP 问题, 很难找到最优化的解, 因此适合采用逐步寻优的方法, 本文采用遗传算法进行求解, 使解不断逼近最优化。

## 2 遗传算法求解组播路由

### 2.1 遗传算法原理

遗传算法 GA (Genetic Algorithm) 是模拟自然界自然选择机制进行搜索寻优的方法, 它模拟生物在染色体层面的各种遗传优化作用而设计人工寻优方法。GA 的本质是一个群体迭代过程, 从一个随机的初始群体出发, 依据生物优胜劣汰的原则, 通过个体之间的竞争、选择、复制、变异等遗传操作, 产生性能更优的下一代群体, 直到满足环境约束的优良个体或合乎具体的应用准则为止。遗传算法的这种特点, 使其很适合解决具有 NP 难度的问题。图 1 给出了遗传算法的一般操作过程。

从图 1 中可以看到, 遗传算法是模拟了自然界生物优胜劣汰历代繁衍的特性, 通过逐代的优化, 使结果不断逼近最优。算法的关键步骤包含了个体的表示即初始群体的产生、选择即优胜个体的判定、交叉即后代产生、变异即

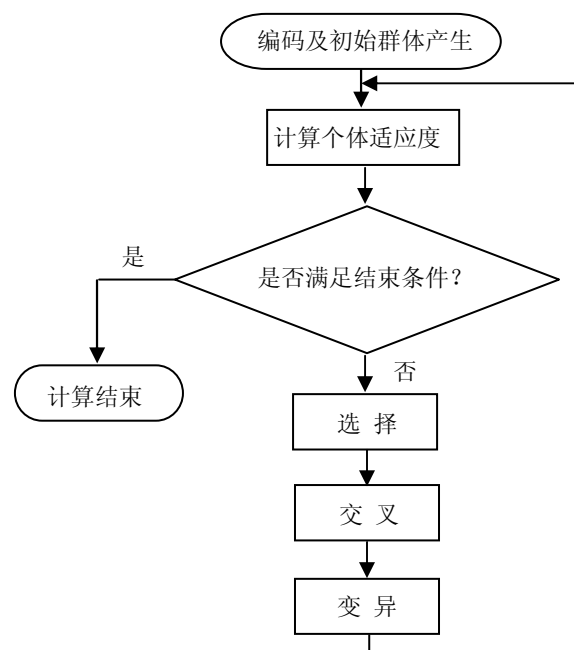


图 1 遗传算法流程图

特殊个体突变产生，完全符合生物的进化过程。接下来，就该组播路由寻优过程，通过改进的遗传算法对其进行说明。

## 2.2 编码

用遗传算法进行求解首先面临的问题是编码，本文采用文献[5]中的编码方法。编码是针对 Steiner 节点进行的。所谓 Steiner 节点，从广义上讲可以包括网络中除了源节点和目标节点的所有其它节点。用二进制位串对其进行编码，位串的长度为  $n-m-2$ ，其中  $n$  和  $m$  分别代表网络中的所有节点数和目标节点数。因此编码的形式如  $\{g(0), g(1), \dots, g(n-m-2)\}$ 。举例来说，对于有八个节点的网络，编号从 1 到 8。其中源节点为 2，目标节点为  $\{3, 5, 7\}$ ，那么遗传算法的编码长度就应该为 4 位， $g(0), g(1), \dots, g(n-m-2)$  分别代表节点 1, 4, 6, 8。若染色体编码为  $\{0, 1, 0, 1\}$ ，则染色体代表的 Steiner 节点集为 4 和 8。

## 2.3 解码

进行遗传算法的时候，必须将编码好的个体按一定的方式进行解码，以便于适应度的计算。在本文中使用的解码算法是建立在经典 KMB 算法的基础之上的。主要分为三步：(1) 使用 Floyd 算法求出图  $G=(V, E)$  的距离图（也称为闭图），得到网络拓扑中的所有节点两两之间的最小路径代价和相应的路径延时。(2) 对距离图执行最小生成树算法，在本文中选取的是 Prim 贪婪算法。(3) 对最小生成树进行扩展，去掉可能产生的环，并且对整棵树进行修剪，删除非目的的叶子节点。

## 2.4 适应度函数

适应度函数是对遗传进化过程中的个体进行评价的客观依据，因此适应度函数的选取就显得格外重要。在本文中选取的适应度评价函数如 (3) 式所示：

$$F(\phi) = \begin{cases} (B-f(\phi)) \times (1 - \frac{d-\Delta}{\Delta}) & (d > \Delta) \\ B-f(\phi) & (d < \Delta) \end{cases} \quad (3)$$

式中  $\phi$  代表遗传群体中的一个个体， $F(\phi)$  为个体  $\phi$  的适应值， $f(\phi)$  则是个体的路由费用， $d$  为相应的组播路由时延， $\Delta$  为实现规定好的路由延时最大边界值， $B$  在本式中为一个相对较大的值，乘号右边的称之为比例因子。需要说明的是，在进化初期，由于个体的分布具有随机性，个体之间的差异也可能较大，计算出来的路由延时可能会比较大，甚至超过了延时最大边界条件的两倍。这时候用上式进行计算会得到负的适应值，因此必须对上式进行修正。并且规定当路由时延很大，超过了最大延时边界条件的两倍时，个体的适应值不再按照 (3) 式计算，直接去乘以一个较小的比例因子，在本文取为 0.01。

分析一下 (3) 式，不难发现该适应度函数考虑了费用和时延这对矛盾。这是因为，当个体满足时延约束时，路由费用越低的个体其适应度就越高，因而在遗传复制中被选择的概率就越大；而当时延超过边界值  $\Delta$  时，说明个体所表示的路由不满足事先规定的条件，可能会对网络中的数据传输造成严重影响，应尽量避免这种情况的发生。此时，比例因子小于 1，使得个体的适应度减小，而且时延超过得越多适应度下降得就越快。这样在遗传操作中就可以尽量地淘汰这些不良个体。

此外，在本文的仿真中，对普通的遗传算法进行了改进。对于遗传算法中最重要的操作交叉，采取了自适应的方法。当遗传进化初期，由于个体之间的差异较大，交叉以较大的概率进行，而到了进化的后期，个体之间的差异不再明显，所以应当适当地减小交叉的概率，使得算法趋于稳定。对选择的策略也进行了相应的改进。在仿真中采取最佳个体直接从父代遗传到子代，其他个

体按照 Z. Michalewicz 提出的非线性排序的选择策略, 将它们按适应从高到低的顺序依次排列, 并按下式分配选择概率:

$$P_i = \begin{cases} q(1-q)^{i-1} & i = 1, 2, \dots, M-1 \\ (1-q)^{M-1} & i = M \end{cases} \quad (4)$$

其中,  $q$  是一个常数, 表示最好的个体的选择概率.

### 3 仿真实验

在本文的仿真中, 为了保证仿真结果更接近实际情况, 按照 Waxman 提出的网络拓扑生成器<sup>[6]</sup> 随机化地生成具有实际网络特性的拓扑模型. 定义一个  $3000 \text{ km} \times 3000 \text{ km}$  的矩形区域作为网络的覆盖范围,  $N$  个网络节点在这个区域内随机分布, 网络节点的距离使用欧拉距离, 任意两节点  $i$  和  $j$  之间的连接边根据节点之间的距离按照一定的概率产生, 边产生的概率按下式决定:

$$p(i, j) = \beta \times \exp\left(\frac{-d(i, j)}{L \times \alpha}\right) \quad (5)$$

式中,  $d(i, j)$  是节点  $i$  和  $j$  之间的距离,  $L$  是两节点之间的最大距离,  $\alpha$  和  $\beta$  是  $(0, 1]$  之间的参数. 当  $\beta$  增大时, 节点间网络边的密度增大; 当  $\alpha$  减小时, 距离短的边相对于距离长的边的密度增大. 在计算路由费用时, 边的距离作为链路费用, 对链路的延时将它定义为边的距离  $\times 1000/v$ ,  $v$  为数据在链路上传输的速度, 取为光速的  $2/3$  倍. 在这里只考虑数据在链路上传输的延时, 忽略信息在交换节点的缓冲区中的排队等待时间.

在仿真中, 按照参与组播通信的网络节点数约为网络节点数的  $15\% \sim 20\%$  的原则, 在生成的网络拓扑中, 对于节点数在  $20 \sim 30$  之间的, 随机选取了 5 个节点构成组播组, 而对于超过 30 个节点的拓扑模型, 本文随机选取了 6 个节点作为组播节点. 时延约束取为  $\Delta = 0.05$ . 路由费用的仿真结果如表 (1) 所示, 路由延时仿真结果如表 (2) 所示. 图 2 是各种算法的路由费用比较, 图 3 是各种算法的路由延时比较.

从图表中不难看出, 用 Dijkstra 算法计算得到的路由时延最小, 但它是牺牲路由费用为代价的, KPP1 算法更注重路由费用的优化, 而 KPP2 算法则结合了路由的时延约束. 从图中的显

表 1 路由费用仿真结果

网络节点数	20	22	24	26	28	30	32	34	36	38	40
Dijkstra 算法	5819	7647.4	5917.7	8362.1	5305.5	11174	7994.3	6893.4	9963.8	11131	14869
KPP1 算法	4343.6	7023.2	5069	7589.4	4262.2	8237.6	6823	7319.1	8811.7	10954	9114.7
KPP2 算法	4388.6	7646.4	5142.9	8362.1	4533.1	8586.3	6823	7387.7	8479.4	10427	9965.4
本文算法	4892.2	6389.3	4891.8	8041.5	4262.2	7359.6	6823	6554.9	8108.7	10140	8546.3

表 2 路由延时仿真结果

网络节点数	20	22	24	26	28	30	32	34	36	38	40
Dijkstra 算法	0.0106	0.0179	0.014	0.0197	0.0122	0.0177	0.0177	0.0152	0.0167	0.0249	0.0167
KPP1 算法	0.0124	0.0295	0.0212	0.0432	0.0213	0.031	0.0261	0.0233	0.0333	0.0390	0.0456
KPP2 算法	0.0124	0.0179	0.019	0.0197	0.0122	0.0299	0.0261	0.0202	0.0225	0.0288	0.0293
本文算法	0.0124	0.0199	0.019	0.0325	0.0213	0.0212	0.0261	0.0192	0.0271	0.0249	0.0311

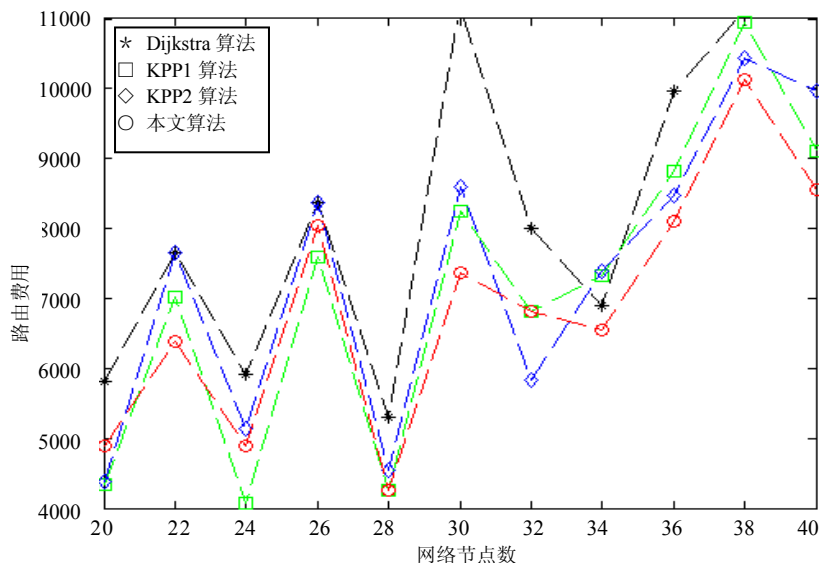


图 2 路由优化遗传算法的路由费用比较

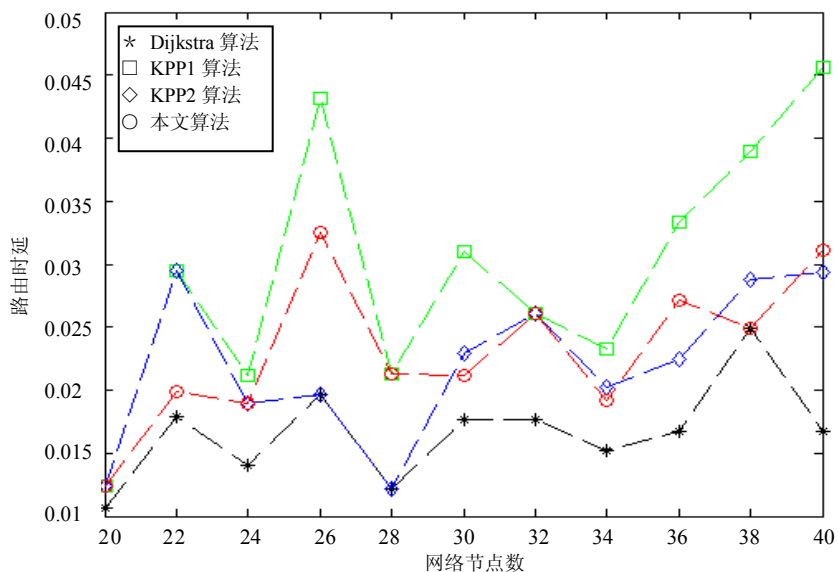


图 3 路由优化遗传算法的路由时延比较

示结果来看，这也符合了费用和时延所表现出来的矛盾的规律，费用的降低将导致时延有所增加。本文提出的改进遗传算法，综合考虑了费用和时延这对矛盾，并在算法中采用了最佳个体保留、自适应交叉以及非线性排序选择等多种优化机制。从仿真结果的曲线来看，路由费用基本上是最小的，而时延和 KPP2 算法差不多。综合优化结果，本文的算法比其他三种算法要好一些。这说明，本文提出的基于遗传进化的路由选择算法是可行的，也是有效的。

参考文献

[1] Kompella V P. Multicast routing for multimedia communication [J]. IEEE/ACM Transaction on Networking, 1993, 1(3): 286-292.

- [2] ZHU Q. A Source-Based Algorithm for Delay-Constraint Minimum-Cost Multicasting [A]. In: Garcia-Luna-Aceves J J. IEEE INFOCOM 95 [C]. Boston: INDEX TERMS, 1995. 377-385.
- [3] Langendoerfer H, Sun Q. Efficient Multicast Routing for Delay-Sensitive Applications [A]. In: Glenford M. The Second Workshop on Protocols for Multimedia Systems PROMS [C]. Salzburg: IEEE, 1995. 452-458.
- [4] Slama H F. Evaluation of multicast routing algorithm for real-time communication on high-speed networks [J]. IEEE Journal on Selected Areas in Communications, 1997, 15(3): 332-345.
- [5] Rampal S, Reeves D S. An evaluation of routing and admission control algorithms for multimedia traffic [J]. Computer Communications, 1995, 18(10): 755-768.
- [6] 严蔚敏. 数据结构[M]. 北京: 清华大学出版社, 1992. 179-185.

## Solving Multicast Routing Problem with Delay Constraint Based on a Modified Genetic Algorithm

CAI Qibo, FANG Jianxiao

(School of Physics and Electronic Information, Wenzhou University, Wenzhou, China 325035)

**Abstract:** In this paper, an improved genetic algorithm is proposed for multicast routing problem, in which both cost and end-to-end delay are generally considered and several optimization methods such as elitist reserved model, self-adaptation crossing and non-linear rank-based model are also adopted. Simulation results shows that this algorithm is feasible and efficient.

**Key words:** Multicast routing; Genetic algorithm; End-to-end delay constraint

(编辑: 王一芳)