

文章编号:1671-9352(2007)09-0041-05

一种解多 QoS 约束组播问题的改进蚁群算法

石钊¹, 葛连升^{1,2*}

(1. 山东大学 计算机科学与技术学院, 山东 济南 250061; 2. 山东大学 网络中心, 山东 济南 250100)

摘要:多 QoS 约束的组播路由技术是当前实现分布式网络多媒体的关键技术,蚁群算法是解决多 QoS 约束组播路由问题的一种启发式算法,但存在收敛速度慢等缺陷. 基于这种情况提出了一种改进的蚁群算法,该算法采用基于组成员节点驱动的方式构造组播树,并在概率转移函数中添加了方向因子,使蚂蚁在寻找路径时摆脱了最初的盲目性,以更大的概率快速向源节点靠近. 仿真实验证明,改进蚁群算法的收敛速度明显提高.

关键词:蚁群算法;组播路由;方向因子;服务质量(QoS)

中图分类号:TP393 **文献标志码:**A

Modified ant colony algorithm for the multi-QoS constraint multicast routing problem

SHI Zhao¹, GE Lian-sheng^{1,2*}

(1. School of Computer Science and Technology, Shandong University, Jinan 250061, Shandong, China;
2. Network Center, Shandong University, Jinan 250100, Shandong, China)

Abstract: The multicast routing technology with multi-QoS constraint is the key for distributed multimedia on the network. The ant colony algorithm is one of the best algorithms for solving the multicast routing problem with multi-QoS constraint, but it takes too much time to converge. Therefore, a modified ant colony algorithm was proposed. This new algorithm adopts the group-member-node-driven method to create the tree and adds the orientation factor to the probability function, which enables the ant to get rid of the initial blindness when searching paths. The simulation results show that the new algorithm can rapidly converge.

Key words: ant colony algorithm; multicast routing; orientation factor; quality of service (QoS)

0 引言

随着网络通信技术和 Internet 的广泛应用,分布式多媒体技术的使用在人们的生活中发挥着越来越大的作用. 作为这些分布式实时应用的关键技术,组播路由技术近年来逐渐成为研究的热点^[1]. 在以往的研究中,组播路由问题常归结为最小费用的组播树问题,即 Steiner Tree₁ 问题,属于树优化问题. 后来人们发现,要实现实时的数据图像传播,必须考虑带宽、时延、丢包率等 QoS 保证,这样的

组播路由问题称为约束 Steiner Tree 问题. 而约束 Steiner Tree 问题已经被证明是一个 NP-完全问题^[2],不存在多项式时间的解决方法,解决此类问题一般采用启发式算法.

在目前已有的启发式算法中,蚁群优化算法以其健壮性、并行性、灵活性以及搜寻过程不需要人工干预和求解精度高的特点得到了广泛应用^[3,4]. 已经有学者将蚁群算法应用于求解 QoS 约束组播问题^[5-7],但这些算法在进行大规模优化时,都存在着初期收敛速度慢、收敛时间过长等缺陷. 本文提出了一种改进的蚁群算法,并将其应用于求解 QoS 约

收稿日期:2007-03-25

基金项目:国家自然科学基金(NSFC-60773101);中国下一代网络 CNGI 大规模路由和组播技术的研究与试验(CNGI-04-13-2T)资助项目

作者简介:石钊(1983-),男,硕士研究生,研究方向:网络路由与组播. Email: shizhao@sdu.edu.cn

* 通讯作者: Email: lsge@sdu.edu.cn

束组播路由问题.仿真结果显示,该算法的收敛速度大大加快.

1 多 QoS 约束组播路由问题的数学模型

通常把通信网看成一个有向连通图 $G(V, E)$, V 表示网络节点集合, E 表示网络链路集合.定义源节点 $s \in V$ 和组成员节点集合 $D \subset V$, 定义集合 T 为图 G 中连接源节点 s 和所有组成员节点 $d_i \in D$ 的所有组播树集合.给定一个组播树 $t \in T$, $e(i, j) \in t$ 是组播树 t 中的一条边, $p_i \in t$ 是组播树中从 s 通往组成员节点 d_i 的路径, 于是可以定义组播树的 QoS 参数如下:

$$\text{bandwidth}(t) = \min\{\text{bandwidth}(e(i, j)), e(i, j) \in t\}, \quad (1)$$

$$\text{delay}(t) = \max\{\text{delay}(p_i), d_i \in D\}, \quad (2)$$

$$\text{delay}(p_i) = \sum_{e(j, k) \in p_i} \text{delay}(e(j, k)), \quad (3)$$

$$\text{loss}(t) = \max\{\text{loss}(p_i), d_i \in D\}, \quad (4)$$

$$\text{loss}(p_i) = 1 - \prod_{e(j, k) \in p_i} (1 - \text{loss}(e(j, k))), \quad (5)$$

$$\text{cost}(t) = \sum_{e(j, k) \in t} \text{cost}(e(j, k)), \quad (6)$$

其中, $\text{bandwidth}(e(i, j))$ 表示链路 $e(i, j)$ 的带宽, $\text{delay}(e(j, k))$ 表示链路 $e(j, k)$ 的时延, $\text{loss}(e(j, k))$ 表示链路 $e(j, k)$ 的丢包率, $\text{cost}(e(j, k))$ 表示链路 $e(j, k)$ 的费用.

基于以上定义,多约束 QoS 组播路由问题可以描述为一个线性规划问题:

$$F = \min\{\text{cost}(t)\}, \quad (7)$$

$$\begin{cases} \text{bandwidth}(t) \geq B_{\min} \\ \text{delay}(t) \leq D_{\max} \\ \text{loss}(t) \leq L_{\max} \end{cases}$$

其中, F 是线性规划的目标函数, B_{\min} 为最小带宽约束, D_{\max} 为最大时延约束, L_{\max} 为最大包丢失率约束.

2 改进蚁群算法

2.1 方向因子的提出

在现实网络中,每台路由器的所在地都有一个惟一的地理位置坐标.因此,可以根据各路由器的地理位置坐标把网络拓扑标注在一张真实的地图上.例如,图1表示的是中国教育科研网骨干网拓扑示意图.

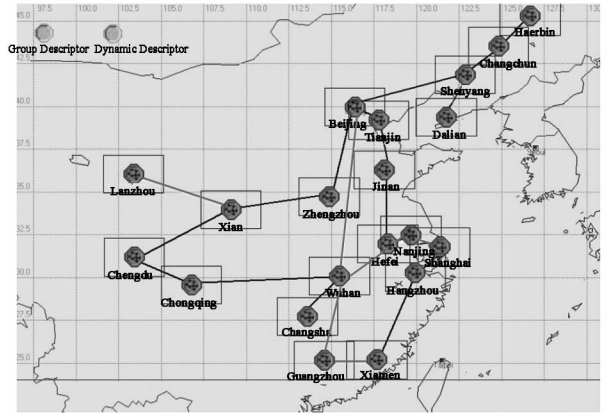


图1 中国下一代互联网示范工程 CNGI-CERNET2 骨干网拓扑图

Fig.1 Backbone topology of China Next Generation Internet (CNGI-CERNET 2)

通过观察图1,发现该拓扑具有如下特点:

(1)骨干网的网络拓扑近似为一张平面图.

(2)网络拓扑中的边与在地理位置上距离比较近的节点连接,而距离比较远的节点则往往通过多条边和多个中间节点连通在一起.

(3)从源节点到目标节点的最优路径往往具有很强的方向性,即该路径上所有的有向边都近似指向目标节点方向,而很少背离该方向.

对于路由器来说,存储自己的地理地址只需要很少的空间,这是完全可行的.同时路由器可以很方便地通过单播的方式将自己的地理地址告知网络上的另一个节点,也可以通过广播的方式将自己的地理地址告诉网络上的所有节点.国外已经有学者做过相关的研究工作^[8,9].因此,本文假定每台路由器都已经存储了自身的地理地址,并具备了地址的分发能力.

在以上的分析和假设的基础上,本文为网络链路添加了一个新的属性,该属性可以用来度量链路指向的方向跟最终要到达的节点的所在方向之间的偏差.将这个新属性命名为方向因子,链路 $e(i, j)$ 的方向因子记为 λ_{ij} , 定义为:

$$\lambda_{ij} = a + \cos\theta, \quad (8)$$

其中, a 是一个常数,且 $a > 1$, θ 是向量 \mathbf{v}_1 与 \mathbf{v}_2 之间的夹角, \mathbf{v}_1 表示从节点 i 所在位置指向它的邻居节点 j 所在位置的方向向量, \mathbf{v}_2 表示从节点 i 所在位置指向参照节点 node_{ref} 所在位置的方向向量,参照节点 node_{ref} 可以是网络上的任意节点.通过向量运算公式 $\cos\theta = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{|\mathbf{v}_1| \cdot |\mathbf{v}_2|}$, 可以方便地计算 $\cos\theta$. 由 λ_{ij} 的定义式可知, $\lambda_{ij} > 0$, 且当 λ_{ij} 的值较大时,沿

链路 $e(i, j)$ 的方向前进可以到达参照节点 node_{ref} 的概率也比较大,当 λ_{ij} 的值较小,沿链路 $e(i, j)$ 的方向前进可以到达参照节点 node_{ref} 的概率也比较小。

2.2 改进蚁群算法的基本操作

跟基于源节点驱动的现有的蚁群算法^[5-7]不同,本文提出的改进蚁群算法是基于组成员节点驱动的.具体过程是:蚂蚁在每个组成员节点处生成,然后开始向源节点迁移,同时记录迁移路径的反向路径信息;当来自所有组成员节点的蚂蚁都到达源节点时,源节点执行路径合并操作,将蚂蚁寻得的各项路径合并成一棵组播树;定期更新链路信息素的浓度.反复进行上述的寻路、路径合并和信息素更新这3个基本操作,直到算法收敛.下面详细介绍本算法中的寻路、路径合并和信息素更新这3个操作。

2.2.1 路径寻找

当蚂蚁在某个组成员节点处产生之后,就开始在网络上迁移,迁移的最终目的地是组播应用的源节点 s . 每一步移动之前,蚂蚁都要按照一定的概率从当前节点的邻居节点中选出一个作为下一跳节点,概率转移函数定义为:

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\lambda_{ij}]^\beta}{\sum_{k \in \text{allowed}} [\tau_{ik}(t)]^\alpha [\lambda_{ik}]^\beta}, & j \in \text{allowed}, \\ 0, & \text{其他}. \end{cases} \quad (9)$$

其中, $p_{ij}(t)$ 表示 t 时刻,在节点 i 上蚂蚁选择节点 j 作为下一跳节点的概率, $\tau_{ij}(t)$ 表示 t 时刻时链路 $e(i, j)$ 上信息素的浓度, λ_{ij} 表示链路 $e(i, j)$ 上方向因子的值,其参照节点就是组播应用的源节点 s . $\text{allowed} = \text{neighbor}_i - \text{tabu}$ 表示下一跳候选节点集合. neighbor_i 表示节点 i 的邻居节点集合, tabu 表示由蚂蚁已经走过的节点、输入链路带宽不满足约束的邻居节点、输入链路延迟不满足约束的邻居节点和输入链路丢包率不满足约束的邻居节点组成的集合, tabu 随着蚂蚁的迁移过程做动态调整. λ_{ij} 可以由每个网络节点自行计算生成,也可以由人工蚂蚁在迁移的过程中计算并一直保存到算法结束.当蚂蚁选择了链路 $e(i, j)$ 之后,记录其反向链路 $e(j, i)$ 及其费用值。

蚂蚁根据概率转移函数公式(9)选择下一跳节点,不断地在网络上迁移.当蚂蚁到达组播应用的源节点 s 时,停止迁移,并将所携带的路径信息交给节点 s . 因为每个组成员节点上的蚂蚁都是相互独立的向源节点移动的,没有先后顺序之分,而每只蚂蚁在选择下一跳时,只用到了节点的局部信息,所以该

蚂蚁寻路操作可以用分布式和并行的方式来实现。

2.2.2 路径合并

当来自所有目标节点的蚂蚁都顺利到达源节点时,我们便得到了从源节点到各个组成员节点的路径 $\text{path}_1, \text{path}_2, \dots, \text{path}_n$ (n 为组成员节点个数),且这些路径都满足时延、带宽和丢包率约束.但是由这些路径生成的子图一般不是一棵组播树,因为生成子图中有可能存在环路,所以将路径合并成树的关键问题是去除环路,下面介绍本文采用的去环算法。

去环算法步骤:

Step1: $i = 1$.

Step2: $j = i + 1$.

Step3: 检测并去除路径 path_i 和 path_j 中可能存在的环路。

Step4: $j++$; 如果 $j \leq n$, 则跳至 Step3.

Step5: $i++$; 如果 $i < n$, 跳至 Step2; 否则算法结束。

完成去除环路操作之后,就得到了不会产生环路的新路径 $\text{path}_1, \text{path}_2, \dots, \text{path}_n$. 由这些新路径生成一个网络子图,该子图不含环路,是 QoS 约束组播路由问题的一个可行解。

2.2.3 信息素更新

为了找到全局最优解,蚂蚁需要定期更新网络链路上信息素的浓度,以形成有用信息的正反馈机制.信息素更新公式如下:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t), \quad (10)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t), \quad (11)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{C_k}, & \text{第 } k \text{ 棵树包含链路 } e(j, i), \\ 0, & \text{其他}. \end{cases} \quad (12)$$

其中, ρ 表示挥发系数,用于控制信息素的挥发速度,且 $\rho \in [0, 1]$, $\Delta\tau_{ij}(t)$ 表示本次循环中链路 $e(i, j)$ 上的信息素增量, $\Delta\tau_{ij}^k(t)$ 表示第 k 棵组播树决定的链路 $e(i, j)$ 上的信息素增量, Q 是常数, C_k 表示第 k 棵树的费用值.初始时刻 $\Delta\tau_{ij}(0) = 0$, $\tau_{ij}(0) = A$, A 为常数。

2.3 改进蚁群算法步骤描述

Step1: 初始化链路信息素浓度和方向因子的值;迭代次数计数器 $\text{iter} = 0$.

Step2: 记录组播树数量的计数器 $\text{count}_r = 0$.

Step3: 每一个组成员节点 d_i ($i = 1, 2, \dots, n$, n 为组成员节点个数)处都产生一只蚂蚁,蚂蚁按照公式(9)定义的概率选择下一跳节点,在网络上不断迁

移.

Step4: 从 d_i 出发的蚂蚁到达源节点 s 后停止迁移, 并将由 s 到 d_i 的路径 $path_i$ 交给 s .

Step5: 当所有组成员节点产生的蚂蚁都顺利到达源节点 s 时, s 运行路径合并算法, 将各个路径 $path_i$ 合并生成一棵组播树 $count_T++$.

Step6: 若 $count_T < MaxTreeNumber$, 则跳转至 Step3.

Step7: 计算各组播树的费用值, 根据公式(10), (11), (12)更新网络链路上的信息素浓度.

Step8: $iter++$, 如果算法已经收敛或者 $iter > MaxIterNumber$, 则算法停止, 输出结果; 否则, 跳转至 Step2.

3 仿真结果和数值分析

通过仿真实验的方式对改进蚁群算法的性能进行测试. 采用设计的基于 Waxman 模型的拓扑生成器^[10]来随机生成网络拓扑, 组播应用的源节点、组成员也通过随机选择方式产生. Waxman 模型中的参数 α 取值为 0.05, β 取值为 2.2. 使用 C++ 语言实现改进蚁群算法(简记为 MACA), 在 P4(1.89 GB) PC 机上运行改进蚁群算法, 并跟以文献[7]中的算法为代表的传统蚁群算法进行比较(简记为 ACA), 得到图 2 所示的实验数据.

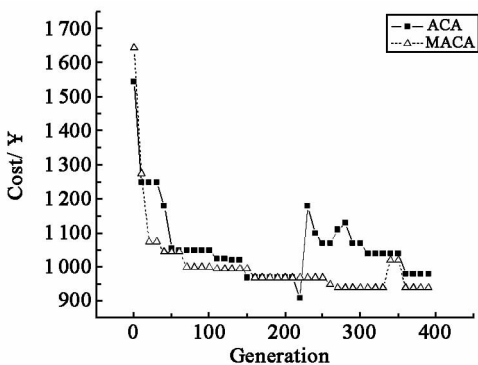


图2 ACA和MACA的收敛情况对比图

Fig.2 Comparison of ACA and MACA in convergence

图2显示在拥有40个节点的随机拓扑上, 改进蚁群算法和传统的蚁群算法的收敛情况比较. 由图中曲线可以看到, 虽然2个算法最终都能收敛到一条性能较好的全局近似最优解上, 但是传统蚁群算法在最终收敛之前的费用值曲线的起伏比较剧烈, 需要运行相当多的迭代次数才能收敛, 而改进后的蚁群算法的费用值曲线起伏则相对平缓, 很快就收敛到了全局近似最优解.

本文又作了扩展性实验, 在不同规模的网络拓

扑上运行仿真程序, 得到了如图3所示的实验数据.

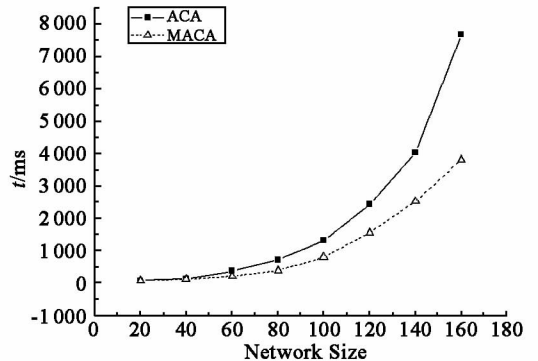


图3 ACA和MACA运行时间对比图

Fig.3 Comparison of ACA and MACA in computing time

由图3可以看到, 改进后蚁群算法的时间曲线一直在改进前的蚁群算法的时间曲线的下方. 这说明, 在不同规模的网络拓扑上, 算法改进后运行的时间消耗都要低于蚁群算法改进前的时间消耗, 且这种趋势随着网络规模的增大而越来越明显. 以上的实验数据证明, 在搜索速度方面, 改进蚁群算法要明显优于传统的蚁群算法.

4 结语

本文将路径的方向性和蚁群算法相结合, 提出了一种改进的 QoS 组播路由算法, 该算法采用了组成员节点驱动的方式构建组播树, 并在概率转移函数中加入了方向因子, 使蚂蚁在寻找路径时摆脱最初的盲目性, 以更大的概率快速地向源节点移动, 较好地克服了蚁群算法中存在的收敛速度慢的缺陷. 仿真实验结果显示, 改进后的算法在收敛速度上要优于传统的蚁群算法. 但是, 该算法没有考虑组成员节点动态变化的情况, 这将是我们的未来研究方向.

参考文献:

- [1] Laxman H Sahasrabudhe, Biswanath Mukherjee. Multicast routing algorithms and protocols; A tutorial[J]. IEEE Network, 2000, 14(1):90-102.
- [2] Vachaspathi P Kompella, Joseph C Pasquale, George C Polyzos. Multicasting for multimedia applications[C]// Proceeding of Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies. Florence, Italy: IEEE Computer Society Press, 1992:2078-2085.
- [3] Marco Dorigo, Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1):53-56.

- [4] Marco Dorigo, Vittorio Maniezzo, Alberto Coloni. Ant system: Optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man and Cybernetics, 1996, 26(1):29-41.
- [5] LU Guoying, LIU Zemin, ZHOU Zheng. Multicast routing based on ant algorithm for delay-bounded and load -balancing traffic[C]// 25th Annual IEEE International Conference on Local Computer Networks. Tampa, Florida: IEEE Computer Society Press, 2000:362-368.
- [6] WANG Ying, XIE Jianying. Ant colony optimization for multicast routing[C]// The 2000 IEEE Asia-Pacific Conference on Circuits and Systems. Tianjin, China: IEEE Computer Society Press, 2000:54-57.
- [7] CHU Chao-Hsien, GU JunHua, HOU Xiang Dan, et al. A heuristic ant algorithm for solving QoS multicast routing problem[C]// Proceedings of the 2002 IEEE Congress on Evolutionary Computation. Honolulu, HI, USA: IEEE Computer Society Press, 2002:1630-1635.
- [8] FADIA A, ANKIT P. Getting geographical information using an IP Address[M]. New York: Association for Computing Machinery, 2000.
- [9] Gene Connolly, George Markowsky, Anatoly Sachenko. Distributed Traceroute Approach to Geographically Locating IP Devices[C]// Proceedings of 2003 Spring IEEE Conference on Technologies for Homeland Security. Boston, USA: IEEE Computer Society Press, 2003:128-131.
- [10] Hussein F Salama, Douglas S Reeves, Yannis Viniotis. Evaluation of multicast routing algorithms for real-time communication on high-speed networks[J]. IEEE Journal on Selected Areas in Communication, 1997, 15(3):332-345.

(编辑:孙培芹)