

分布式蚁群 QoS 组播路由算法

许 洪, 王 华, 伊善文

(山东大学计算机科学与技术学院, 济南 250101)

摘 要: 为寻找满足带宽、延迟等约束的最小代价组播路由树的 QoS 组播路由, 提出一种分布式蚁群优化算法。在蚂蚁行动过程中形成组播树的森林。若在某步结束时森林中仅剩 1 棵树, 则蚂蚁达到目标, 停止行动。仿真实验结果表明, 该算法在结果和收敛速度方面都有良好的表现。

关键词: 服务质量; 组播路由; 蚁群优化算法; 分布式实现

Distributed Ant Colony QoS Multicast Routing Algorithm

XU Hong, WANG Hua, YI Shan-wen

(School of Computer Science and Technology, Shandong University, Jinan 250101)

【Abstract】 According to the QoS multicast routing problem, which is to find the least cost multicast routing tree that satisfies bandwidth, delay and other constraints, this paper proposes a distributed ant colony optimization algorithm. In this algorithm, a multicast tree's forest is formed during the ant movement progress. The objective of ant movement is to connect all the trees in the forest to form a single tree. The most important advantage of the algorithm is distributed implementation. Simulation experiments show that this algorithm performs well both in results and convergence speed.

【Key words】 QoS; multicast routing; ant colony optimization algorithm; distributed implementation

1 概述

网络多媒体业务如计算机会议、远程教学等正逐渐变成互联网的主流业务, 这些业务采用组播解决是十分方便的, 但这些应用往往对带宽、延迟、抖动、丢失率等 QoS 参数有严格的要求。

QoS 组播路由的核心问题是建立满足特定 QoS 参数要求的组播源根树, 通常可以形式化为图论中的 Steiner 树问题, 通过求解 Steiner 最小树来求解代价最小的组播树。由于该问题是 NP 完全问题^[1] 因此许多解决该问题的精确算法的计算量都是随图中节点的增加而指数增加, 并不适用于大规模应用; 而像 KMB^[2]和 SPH^[3]等启发式算法虽然能够快速地找到一棵费用接近最优的准 Steiner 树, 但它们并不能得到较好的解。近年来随着禁忌搜索、遗传算法和蚁群算法等元启发算法的兴起, 不少学者将这类算法运用到组播的求解中^[4-5], 并且取得了一定的成果。各算法都有各自的优缺点, 其中蚁群优化算法以其健壮性、灵活性、搜寻过程不需要人工干预以及求解精度高的特点, 得到了广泛应用。

综合考虑已有算法的优缺点, 本文提出一种分布式蚁群优化算法来解决 QoS 组播路由问题。

2 组播路由的定义

网络模型可以看成无向连通图 $G(V, E)$, 其中, V 是节点集合(网络节点); E 是边集(网络链路); $|V|$ 为网络节点个数; $|E|$ 为网络链路个数。在链路上面定义 4 个参数: 费用 $C(e)$, 延迟 $D(e)$, 可用带宽 $B(e)$ 和丢包率 $L(e)$ 。链路上的延迟、可用带宽和丢包率定义了必须被限制的标准。

组播树 $T(s, M)$ 是图 G 的一个生成子树, 且这棵子树覆盖了源节点 $s \in V$ 和目的节点集 $M \subseteq V - \{s\}$ 。定义 M 为目的节

点集合; $\{\{s\} \cup M\}$ 为组播组集合; $P_T(s, d)$ 是组播树 T 中从源节点 s 到某一个目的节点 $d \in M$ 的单一路径。

树 $T(s, M)$ 的总费用定义为树上所有链路的费用总和:

$$C(T(s, M)) = \sum_{e \in T(s, M)} C(e) \quad (1)$$

路径 $P_T(s, d)$ 的总延迟简单地定义为路径 $P_T(s, d)$ 上所有链路的延迟之和:

$$D(P_T(s, d)) = \sum_{e \in P_T(s, d)} D(e) \quad (2)$$

路径 $P_T(s, d)$ 的瓶颈带宽定义为路径上所有链路的最小可用带宽值:

$$B(P_T(s, d)) = \min\{B(e), e \in P_T(s, d)\} \quad (3)$$

路径 $P_T(s, d)$ 的总的丢包率定义为用 1 减去路径 p 上所有链路的数据到达率的乘积:

$$L(P_T(s, d)) = 1 - \prod_{e \in P_T(s, d)} (1 - L(e)) \quad (4)$$

树 $T(s, M)$ 的延迟抖动定义为源到各目的节点路径上的延迟的平均差:

$$DJ(T(s, M)) = \sqrt{\sum_{d \in M} (D(P_T(s, d)) - delay_avg)^2} \quad (5)$$

其中, $delay_avg$ 表示源到各目的节点的路径上延迟的均值。

则满足多 QoS 约束的费用最小的组播路由问题可定义为 $\text{Min } C(T(s, M))$

基金项目: 国家自然科学基金资助项目“大规模组播聚合模型及算法研究”(60773101); 下一代网络示范工程基金资助项目“大规模路由和组播技术的研究与试验”(CNGI-04-13-2T)

作者简介: 许洪(1986-), 女, 硕士研究生, 主研方向: 最优化理论, 组播路由; 王华, 副教授、博士后; 伊善文, 硕士研究生

收稿日期: 2009-10-08 **E-mail:** xuhong@sdu.edu.cn

$$\begin{cases} D(P_T(s,d)) & \Delta_d, \forall d \in M \\ B(P_T(s,d)) & B_d, \forall d \in M \\ L(P_T(s,d)) & L_d, \forall d \in M \\ DJ(T(s,M)) & DJ_d \end{cases} \quad (6)$$

其中, Δ_d 为延迟上限; B_d 为带宽下限; L_d 为丢包率上限; DJ_d 为延迟抖动上限。

3 分布式实现的蚁群优化算法

传统的蚁群算法都要借助于蚂蚁对从源到各个目的节点路径的寻找, 然后再把这些路径合并成一棵组播树。这种方式虽然同样有其并行性和分布式实现的优点, 但由于重复性的工作太多, 因此效率不高。基于上述现状, 本文提出一种分布式蚁群优化算法(Distributed Ant Colony Optimization algorithm, DACO)。DACO 既能并行性和分布式实现, 又能以较快的速度收敛到理想的结果。

3.1 DACO 的基本操作

3.1.1 预剪枝

(1)用深度或宽度优先搜索算法遍历网络拓扑图, 遍历的同时去除不属于组播集合的叶子节点以及与之相关联的边, 同时去除那些带宽不符合要求的边。

(2)若预剪枝后的网络拓扑图为非连通图, 则不存在符合 QoS 约束的组播树, 算法结束。

3.1.2 森林的生长

(1)创建一个森林 $F(T_i(E_i, V_i))$, 其中, $T_i(E_i, V_i)$ 为多播树; i 属于组播组集合。初始时设定 $E_i = \text{NULL}$, $V_i = \{i\}$ 为森林中的每一棵多播树创建一个候选链路集合 E_i' , 初始时设定 $E_i' = \{e(i, j) | \text{链路 } e(i, j) \text{ 属于该网络}\}$ 。

(2)为森林中的每一棵树按下式从各自的候选集合 E_i' 中选择一条链路:

$$P_j = \begin{cases} \frac{[\tau_j]^\alpha [\lambda_j]^\beta}{\sum_{e_k \in E_i'} [\tau_k]^\alpha [\lambda_k]^\beta} & e_j \in E_i' \\ 0 & \text{others} \end{cases} \quad (7)$$

其中, e_j 是 E_i' 中的第 j 条链路; τ_j 为 e_j 上的信息素的浓度; λ_j 为 e_j 上的启发式函数, 根据实际情况可以是 e_j 的费用、延迟或丢包率等的函数, 因为本文实验是在满足多约束情况下求树的费用最小, 所以本文实验时选择的是 $\lambda_j = \frac{1}{C(e_j)}$; α 和 β 分别是用来调节信息素和启发式函数所起作用的参数。

(3)假设选出的边为 $e(j, k)$, 执行以下操作:

1)将链路 $e(j, k)$ 加入到 $T_i(E_i, V_i)$ 中的链路集合 E_i 中, 将节点 k 加入到 V_i 中, 即 $E_i = E_i + e(j, k)$, $V_i = V_i + \{k\}$ 。

2)修改集合 E_i' 的内容, 令 $E_1 = \{e(m, n) | e(m, n) \notin E_i, m \in V_i, n \in V_i\}$, $E_2 = \{e(m, n) | m \in V_i, n \notin V_i\}$, 则 $E_i' = E_i' - E_1 + E_2$ 。

由此可见, 每次都选择一条链路加入树 T_i 中, 相当于 T_i 在不断地生长, 且生长速度为 1; 若运行过程中, 某棵树选择的一条边的顶点在另一棵树上, 则两棵树合并为一棵树; 反复执行森林的生长过程; 若某一步结束时森林中只剩下一棵树则森林停止生长。

3.1.3 树的剪枝

由 3.1.2 节找到的树虽然覆盖了所有的组播组集合中的节点, 但仍然不是一棵真正的组播树, 因为它包含了一些非目的节点的叶子节点, 所以需要对其 T 进行剪枝操作, 除去那

些非目的节点的叶子节点以及与之相关联的边。

3.1.4 组播树是否满足 QoS 约束的判断

由于在预剪枝步骤已经判断过网络拓扑上的边是否满足带宽约束, 因此此处就不需要再次判断。

根据式(2)、式(4)~式(6)判断组播树是否满足延迟、丢包率和延迟抖动的约束。若满足, 则计算组播树的费用进入下一步信息素的更新; 否则组播树的费用赋值为 0 进入下一步。

3.1.5 信息素的更新

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。本算法逐代放出蚂蚁, 每代放出蚂蚁个数为 $antnum$ 。信息素更新时同时更新每代蚂蚁所得到的最优树与全局最优树。这样的更新方式可以使我们的算法, 在保证收敛性的同时, 保证寻树的随即性, 使算法得到最优树的可能性增大。

信息素更新公式如下:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij} \quad (8)$$

其中, ρ 表示挥发系数, 用于控制信息素的挥发速度, 且 $\rho \in [0, 1]$; $\Delta \tau_{ij}$ 表示本次循环中链路 $e(i, j)$ 上由局部最优树和全局最优树综合得出的信息素增量。

重复以上各操作, 直至算法收敛或者达到最大迭代次数; 输出最优解, 算法结束。

3.2 DACO 空间和时间复杂度分析

(1)空间复杂度分析

由于存储网络拓扑图和信息素的是邻接矩阵, 而所有蚂蚁共享这个矩阵, 因此需要的空间大约为 $O(|V|^2)$ 。

(2)时间复杂度分析

共有 $antnum$ 只蚂蚁共同找树, 所以总的时间复杂度可记为 $O(antnum|E||V|)$ 。又由于算法在迭代多次之后满足收敛条件时结束, 因此总的时间复杂度可记为 $O(iter \times antnum|E||V|)$ 。

4 仿真实验与结果分析

本文选择了由美国北卡罗莱纳州立大学高级计算通信中心的 Salama 等人开发的 MRSIM。该平台使用 C++ 编写, 并给出了接口以实现新研究的算法, 本文在 MRSIM 中实现了本算法和其他相关算法, 并与其他算法进行性能对比。本实验的拓扑是由拓扑生成器 $nem^{[6]}$ 生成的, 如图 1 所示。

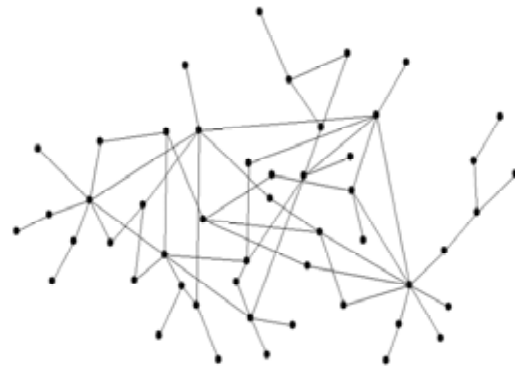


图 1 nem 生成的拓扑图

4.1 预剪枝操作对算法的影响实验

本文对网络拓扑图进行预剪枝的目的是去除那些不属于组播组集合的叶子节点以及与之相关联的边, 同时去除那些带宽不符合要求的边。图 2 即为对图 1 预剪枝之后的拓扑图, 黑色的圆圈为组播组成员。可以看出, 对网络拓扑图进行时间复杂度仅为 $O(|E|)$ 的预剪枝操作之后, 既能明显减少算法

搜索无用边的时间，加快算法收敛的速度，提高算法搜索到最优解的可能性，又能保证算法得到的解优于不进行此操作时算法所得到的解。

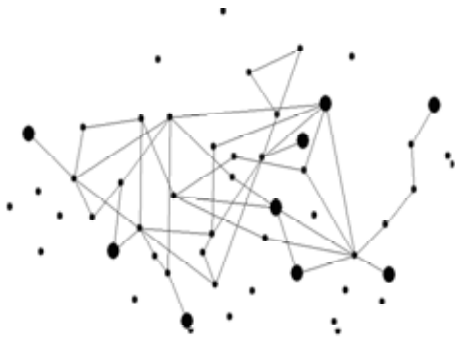


图2 对图1预剪枝后的拓扑图

图3为算法运行预剪枝操作与不运行预剪枝操作算法的收敛时间比较。

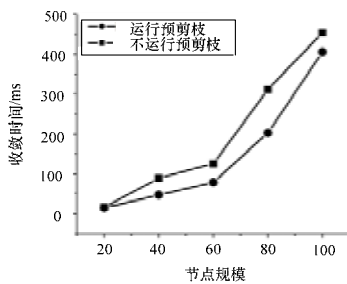


图3 算法运行预剪枝操作与不运行预剪枝操作的收敛时间比较

4.2 仿真实验

为进一步验证 DACO 的性能，DACO 将与文献[5]提出的用传统的蚁群算法解组播的算法、文献[7]提出的解决网络路由问题的 ACO 算法：ANTNET 和文献[4]提出的遗传算法进行各方面性能的比较。ANTNET 算法本身是针对于单播路由提出的算法，但通过修改信息素的更新机制，可以很简单地运用到组播问题的求解中。

各算法是在满足不同的多 QoS 约束时提出来的。为了使各算法有可比性，本文放宽它们的 QoS 约束，即对于这些约束，算法运行时并不会受到它们的限制，从而使得各算法的结果有可比性。

由 nem 生成的随机无向拓扑图，拓扑规模为 20 个~200 个节点，组播源和目的节点随机选定，在目的节点个数占整个拓扑节点总数的 15% 的情况下，4 种算法所得到的组播树费用、收敛时间的比较如图 4~图 5 所示。

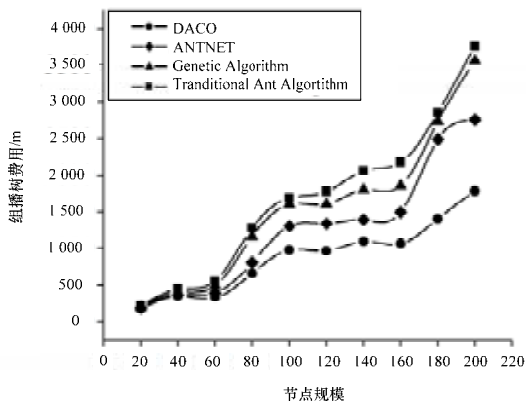


图4 节点规模不同时4种算法所得组播树费用比较

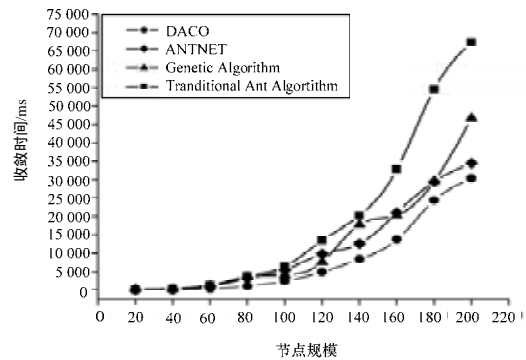


图5 节点规模不同时4种算法的收敛时间比较

可见，本算法所得到的组播树的费用、收敛时间始终优于上述3种算法，且拓扑规模越大，优秀性越明显。这是由于本算法采用森林中的组播组集中的节点逐渐生长成树，再合并成一棵树的方式来生成组播树；与传统的先找端到端最短距离再合并树的方式相比，DACO 选择对组播树全局费用更优的链路的机会更大，并且在信息素作用的影响下多个目标节点共享此链路的可能性也增大，可能获得更优的费用结果，得到一棵多播树的时间复杂度仅为 $O(|E||V|)$ ；而遗传算法初始群体的生成是随机的，交叉、变异和选择算子不能及时利用拓扑的反馈信息，减小了算法找到最优解的可能性，而且交叉都涉及到去环、合并树、剪枝等过程，当拓扑规模增大时，这些过程更复杂，需要耗费的时间更多；所以，相对于传统蚁群算法、ANTNET 和遗传算法，本算法所得组播树的费用、时间始终优于上述3种算法。

5 结束语

本文提出一种可以分布式蚁群优化算法解决 QoS 组播路由问题。算法考虑了多种 QoS 约束条件，包括带宽、延迟、延迟抖动和丢包率。算法最为重要的一个优点是可以分布式实现。仿真结果表明算法在寻优能力和收敛速度方面都优于 ANTNET、传统蚁群和遗传等算法，而且拓扑规模越大，优秀性越明显。在以后的工作中，将进一步研究算法的实际应用，将算法运用于网络协议体系结构，提高算法的实用性。

参考文献

- [1] Zheng Wang, Crowcroft J. Quality of Service for Supporting Multimedia Application[J]. IEEE Journal on Selected Areas in Communication, 1996, 14(7): 1228-1234.
- [2] Kou L, Markowsky G, Berman L. A Fast Algorithm for Steiner Trees[J]. Acta Informatica, 1981, 15(2): 141-145.
- [3] Takahashi H. An Approximate Solution for the Steiner Problem in Graphs[J]. Math Japan, 1980, 24: 573-577.
- [4] Bao Guangbin, Yuan Zhanting, Zhang Qiuyu, et al. A Novel Genetic Algorithm to Optimize QoS Multicast Routing[J]. Lecture Notes in Control and Information Sciences, 2006, 344: 150-157.
- [5] Gong Bencan, Li Layuan, Wang Xiangli, et al. A Novel QoS Multicast Routing Algorithm Based on Ant Algorithm[C]//Proc. of WiCOM'07. Chengdu, China: [s. n.], 2007: 2025-2028.
- [6] Magoni D. nem: A Software for Network Topology Analysis and Modeling[C]//Proc. of MASCOTS'02. [S. l.]: IEEE Computer Society, 2002: 364-371.
- [7] Caro G D, Dorigo M. AntNet: A Mobile Agents for Adaptive Routing[C]//Proc. of HICSS'98. [S. l.]: IEEE Press, 1998.

编辑 金胡考