

蚁群算法在 QoS 网络路由中的应用

孙力娟,王良俊

(南京邮电学院 计算机科学与技术系,江苏 南京 210003)

(sunlj@njupt.edu.cn)

摘 要:采用蚁群算法求解包含延迟、延迟抖动、带宽、丢包率和最小花费等约束条件在内的 QoS 单播路由问题。实验仿真表明该算法能快速有效地求得路由最优解。

关键词:蚁群算法;QoS;网络路由

中图分类号:TP393.07 **文献标识码:**A

Application of ant colony algorithm for solving QoS routing problem

SUN Li-juan, WANG Liang-jun

(Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications,
Nanjing Jiangsu 210003, China)

Abstract: Ant colony algorithm was used to solve the QoS point-to-point routing problem with delay, delay jitter, bandwidth, packet loss and the least cost constraints. Simulation results show that this algorithm is reasonable and effective for solving QoS routing problem.

Key words: ant colony algorithm; QoS; routing

0 引言

随着 Internet 网络应用越来越广泛,先进的多媒体业务层出不穷,各种服务质量(QoS)业务应运而生。QoS 路由的任务就是在网络中寻找一条路由,满足用户对线路的带宽、延时、延时抖动、费用的要求,即向用户提供端到端的服务质量保证。

基于多个不相关可加度量的 QoS 路由问题是 NP 完全问题,目前采用的方法多为启发式算法,如遗传算法、模拟退火算法等,但是在 QoS 路由问题上它们都有自身难以克服的缺陷,要么太复杂,要么太费时。

20 世纪 90 年代,意大利学者 M. Dorigo, V. Maniezzo 等人通过模拟自然界蚂蚁寻径的行为,提出了一种全新的启发式算法——蚁群算法(Ant Colony Algorithm, ACA),较强的鲁棒性、寻径过程的并行性以及易于与其他启发算法结合的优越性,使得蚁群算法被广泛用于解决各种具有 NP 难度的问题。

本文提出了一种基于蚁群算法并考虑多个路由限制的优化 QoS 路由问题的方法。实验仿真表明:该算法能有效地求解 QoS 路由问题。

1 蚁群算法(ACA)的基本原理

蚁群算法是在研究自然界中蚁群行为的基础上提出的一种启发式方法。实验观察表明,在蚂蚁寻找食物时,它们总能找到一条从食物到蚁巢的最优路径。原因是,蚂蚁在运动过程中会在路径上释放出一种特殊的信息素,该物质随着时间的推移会逐渐挥发消失。后面的蚂蚁可根据前面走过的蚂蚁遗留下来的信息素值来选择下一步要走的路径。一条路径上

的信息素值越高,蚂蚁选择这条路径的概率就越大,构成一个学习信息的正反馈过程,最优路径上的信息素浓度越来越大。虽然单个蚂蚁的选路能力有限,但是通过个体之间的信息交流(正反馈),整个蚁群之间不断交换路径信息,最终找出最优路径。

在 ACA 中,每个蚂蚁根据状态转移规则来选择下一跳节点,使用全局更新和局部更新两种规则可以取得更快的全局优化结果。下面介绍这三种规则^[1]:

1) 信息素的全局更新规则

当 m 只蚂蚁成功地完成一次寻径行为后,选择出目标数值最小的一个路由,用来完成全局信息素的更新。以使得较优解保留下来,对后继蚂蚁的行为产生影响,达到较快收敛于最优解的目的。若 r, s 是两个相邻的节点,则有:

$$\tau(r, s) \leftarrow (1 - \alpha)\tau(r, s) + \alpha \times \Delta\tau(r, s) \quad (1)$$

式中:

$$\Delta\tau(r, s) = \begin{cases} (Q/L_{best}), & \text{if } (r, s) \in \text{global-best-tour} \\ 0, & \text{otherwise} \end{cases}$$

式中, $0 < \alpha < 1$, L_k 是目标函数, L_{best} 是最佳目标函数值, global-best-tour 是取得 L_{best} 值的蚂蚁路由, Q 为常数。

2) 信息素的局部更新规则

对于第 k 只蚂蚁,如果节点 r, s 是它所选择路径上的两个相邻节点,则有:

$$\tau(r, s) \leftarrow (1 - \rho) \times \tau(r, s) + \rho \times \Delta\tau(r, s) \quad (2)$$

否则,不更新。其中, $0 < \rho < 1$ 。在文献[3]中:ACA 算法取 $\Delta\tau(r, s) = \tau_0$, τ_0 是各条链路上信息素的初始值。Ant-Q 算法采用 $\Delta\tau(r, s) = \gamma \times \max_{z \in J_k(s)} \tau(s, z)$, 虽然 ACA 算法在局部信息素更新上计算量小于 Ant-Q 算法,但仍可以取得相近的

收稿日期:2004-03-24;修订日期:2004-05-09

作者简介:孙力娟(1963-),女,江苏南京人,副教授,博士研究生,主要研究方向:智能优化方法、计算机网络;王良俊(1979-),女,山东滨州人,硕士研究生,主要研究方向:计算机网络。

结果。因此,本文实验采用 ACA 算法,取 $\Delta\tau(r, s) = \tau_0$, 设为常数 $const$ 。

3) 状态转移规则

在节点 r 的第 k 个蚂蚁选择下一节点 s 的规则是:

$$s = \begin{cases} \arg \max \{ [\tau(r, u)]^\alpha \times [\eta(r, u)]^\beta \}_{u \in J_k(r)}, & \text{if } q \leq q_0 \\ S, & \text{otherwise} \end{cases} \quad (3)$$

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, u)]^\alpha \times [\eta(r, u)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)]^\alpha \times [\eta(r, u)]^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

其中, $\tau(r, s)$ 是两节点 r, s 间的信息素值; $\beta (\beta > 0)$ 是能见度的相对重要性; $J_k(r) = \{0, 1, 2, \dots, n-1\}$ 表示第 k 个蚂蚁下一步可以选择的节点集; $\eta(r, s) = \frac{1}{cost(r, s)}$, $cost(r, s)$ 表示节点 r, s 间的费用大小。

2 基于 ACA 的 QoS 路由优化算法

2.1 QoS 路由由优化问题的描述

QoS 路由的任务是向用户提供端到端的服务质量保证。它的质量指标,包括延时 (Delay)、延时抖动 (Delay_jitter)、线路带宽 (Bandwidth)、丢包率 (Packet_loss) 和费用 (Cost) 等,这些因素构成了 QoS 路由问题的约束条件。本文的路由算法,在满足 4 个约束条件下,求花费最小的路由。

网络模型表示为赋权图 $G = (V, E)$, 其中 V 是图中所有网络节点 (交换机、路由器和主机等) 组成的集合, E 是网络双向链路的集合, 每一条边表示两节点间的直通信路, 假设网络是对称的。 $S \in V$ 为源点, $M \in \{V - \{S\}\}$ 为终点。对于任一链路 $e \in E$, 定义四种属性: 延迟函数 $delay(e)$, 费用函数 $cost(e)$, 带宽函数 $bandwidth(e)$ 和延迟抖动函数 $delay_jitter(e)$ 。对于任一网络节点 $n \in V$, 也定义四种属性, 分别为延迟函数 $delay(n)$, 费用函数 $cost(n)$, 延迟抖动函数 $delay_jitter(n)$ 和包丢失率函数 $packet_loss(n)$ 。则对于给定的源点 $S \in V$, 终点 M, S 和 M 组成的单播树 $T(S, M)$ 存在下列关系:

$$\begin{aligned} delay(T(S, M)) &= \sum_{e \in T(S, M)} delay(e) + \sum_{n \in T(S, M)} delay(n) \\ cost(T(S, M)) &= \sum_{e \in T(S, M)} cost(e) + \sum_{n \in T(S, M)} cost(n) \\ bandwidth(T(S, M)) &= \min \{ bandwidth(e), e \in T(S, M) \} \\ delay_jitter(T(S, M)) &= \sum_{e \in T(S, M)} delay_jitter(e) + \sum_{n \in T(S, M)} delay_jitter(n) \\ packet_loss(T(S, M)) &= 1 - \prod_{n \in T(S, M)} (1 - packet_loss(n)) \end{aligned}$$

QoS 单播路由问题就是在网络 $N(V, E)$ 中, 给定源点 S , 终点 M , 寻找一条路由 $T(S, M)$ 满足:

- 1) 延迟约束 $delay(T(S, M)) \leq D$;
- 2) 带宽约束 $bandwidth(T(S, M)) \geq B$;
- 3) 延迟抖动约束 $delay_jitter(T(S, M)) \leq DJ$;
- 4) 包丢失率约束 $packet_loss(T(S, M)) \leq PL$;
- 5) 费用约束, 在满足前面 4 个约束的条件下, $cost(T(S, M))$ 最小。

其中, B, D, DJ 和 PL 分别代表业务对网络带宽、延迟、延

迟抖动和包丢失率的约束限制。本文的 QoS 路由包含四个互不相关的限制条件, 是一个 NP 完全问题。

2.2 基于 ACA 的 QoS 路由算法

在进行网络路由之前, 需要对网络拓扑结构进行精简, 以简化算法设计的难度, 同时优化算法的性能。本文中除去不满足带宽约束的链路, 把网络滤成一个新的网络, 如果源点和终点都位于同一连通分量, 就将此连通分量作为算法研究的基图。(以下讨论只考虑延迟、延迟抖动和费用。若网络不能满足带宽的约束要求, 可放宽带宽约束重新处理)。

QoS 限制的引入是在全局信息素更新的时候, 即用是否满足限制条件来确定惩罚因子的数值, 调整目标函数值, 从而达到调整 $\tau_{ij}(t + \Delta t)$ 函数大小的目的。在 m 只蚂蚁完成一次寻径后, 计算它们的目标函数 L_k [8]:

$$L_k = \frac{1}{cost[T(r, s)]} (Af_d + Bf_{dj} + Cf_{pl}) \quad (5)$$

$$f_d = \Phi_d \{ delay[T_k(r, s)] - D \}$$

$$f_{dj} = \Phi_{dj} \{ delay_jitter[T_k(r, s)] - DJ \}$$

$$f_{pl} = \Phi_{pl} \{ packet_loss[T_k(r, s)] - PL \}$$

$$\Phi_d(Z) = \begin{cases} 1, & Z \leq 0 \\ r_d, & Z > 0 \end{cases}$$

$$\Phi_{dj}(Z) = \begin{cases} 1, & Z \leq 0 \\ r_{dj}, & Z > 0 \end{cases}$$

$$\Phi_{pl}(Z) = \begin{cases} 1, & Z \leq 0 \\ r_{pl}, & Z > 0 \end{cases}$$

其中, A, B, C 分别为 f_d, f_{dj}, f_{pl} 的正加权函数, 分别表示延迟、延迟抖动和包丢失率在目标函数中所占的比重, 其值可根据具体应用设定。

设 $\Phi_d(Z)$ 是延迟度量的惩罚函数, 当该路由满足延迟约束 ($delay(T_k(r, s)) \leq D$) 时, 值为 1; 否则等于 r_d ($0 < r_d < 1$)。同理, $\Phi_{dj}(Z)$ 是延迟抖动度量的惩罚函数, 当路由满足延迟抖动约束 ($delay_jitter(T_k(r, s)) \leq DJ$) 时, 其值为 1, 否则为 r_{dj} ($0 < r_{dj} < 1$); $\Phi_{pl}(Z)$ 是包丢失率度量的惩罚函数, 当路由满足包丢失率约束时, 值为 1, 否则等于 r_{pl} ($0 < r_{pl} < 1$)。 r_d, r_{dj}, r_{pl} 的值大小决定惩罚的程度, 在本实验中, 选择 $r_d = r_{dj} = r_{pl} = 0.5$ 。比较 L_k ($1 \leq k \leq m$) 大小, 取其值最大者为 L_{best} , 根据公式 (1) 调整更新信息素值的大小。

本文采用 MMAS (Max-Min Ant System) [6] 对基本蚁群算法进行了改进, 算法的详细步骤如下:

1) 初始化参数。假定网络中有 m 只蚂蚁, n 个节点, 给出各个节点的 (d, dj, pl, c) 的取值, 以及每条存在边的 (d, dj, b, c) 的取值, 给出约束条件的 D, DJ, B, PL 的值。

2) 精简网络。

3) 初始化网络拓扑中各条链路的相应信息素值 $\tau_0 = const$, 将 m 只蚂蚁放置在源节点 S 上。

4) 各个蚂蚁依照状态转移规则公式 (3)、(4) 选择下一跳节点。当一只蚂蚁成功地完成下一跳路由选择后, 这只蚂蚁所选择的路径上的信息素将根据局部信息素更新规则公式 (2) 进行更新。若蚂蚁在达到目的节点前找不到合适的下一跳节点, 则宣布这个蚂蚁死亡。

5) 对所有的蚂蚁进行步骤 4), 直到所有的蚂蚁完成步骤 4), 即找到从源点到目的节点的路由或者死亡;

6) 计算目标函数 L_k 的值, 选择 L_k 的最大值 L_{best} , 得到全局最优的路由 T_{best} , 使用全局更新规则对所有网络链路上的

信息素进行更新;

7) 重复步骤4)~6),直到满足结束条件。

3 实验仿真及结果

本文通过编码实现上述蚁群算法(ACA)在 QoS 单播路由问题中的应用,程序用标准的 C 语言编写,共有 708 行。实验采用图 1 所示的网络拓扑,图中每个顶点用 $\langle d, dj, pl, c \rangle$ 表示,其中的元素分别代表节点延时、延时抖动、丢失率和费用;每条边用 $\langle d, dj, b, c \rangle$ 表示,其中的元素分别代表链路延迟、延迟抖动、边的带宽和费用。

蚂蚁数目越多,群体优势就越能得到很好的体现;但是,蚂蚁数目过多会增加系统的计算量。

实验中的仿真参数设置如下:蚂蚁数 = 30, 寻径代数 = 20。初始取 $D = 15, DJ = 12, PL = 0.1, B = 60$, 然后对其值进行适当的调整。各条链路上信息素的初始值,我们设为统一大小,经过多次试验得到: $\tau_0 = 10, Q = 1000$ 。采用 MMAS 算法对 τ 进行限制, τ 在 $[\tau_{\min}, \tau_{\max}]$ 中取值, τ_{\min}, τ_{\max} 的大小由文献[6]取得。

在对算法进行实验前,按照给定的带宽限制 B 对图 1 的网络拓扑进行简化。

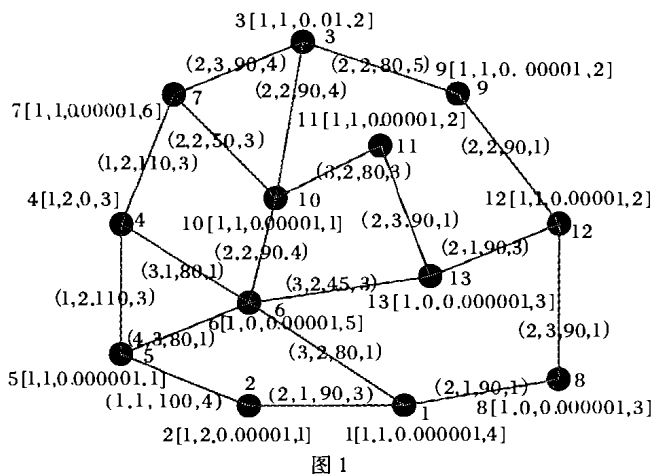


图 1

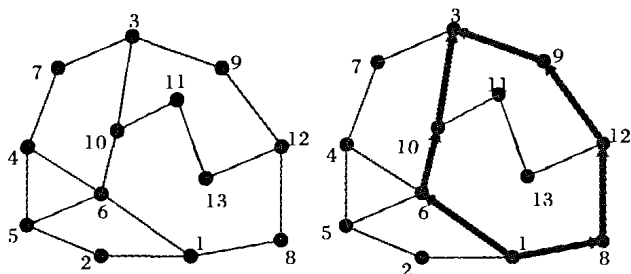


图 2

先由 B 简化网络拓扑图,得到图 2,以下的实验数据由图 2 计算得到。设源点为 1,目的节点为 3, QoS 需求决定加权系数 A, B, C 的大小。本实验验证的是算法的收敛性,所以设定 $A = B = C = 1$,即在满足 QoS 限制下,考虑费用函数的制约作用。

经过 30 只蚂蚁 20 代的寻径,得到的最佳路由如图 3 所示。图中有两条符合 QoS 要求的最优路径,如表 1 所示。我们对成功的路由请求定义如下:算法构成的单播树满足本文研究的 QoS 约束,即满足延时、延时抖动、包丢失率和带宽约束。而其他寻径未成功路径的出现起到了探索新路由的作用,以防止蚂蚁陷入局部最优解。

表 1 (1,3)最佳路由结果

(1,3) 路径	延迟 /s	延迟抖动 /s	丢包率	费用
1-6-10-3	9	7	0.01001	17
1-8-12-9-3	11	10	0.01002	17

表 2 为 4 个参数不同取值情况下的寻径成功率,可见参数的取值对算法的收敛性起决定作用。这些参数如何取值进行组合,才能达到算法的最优结果,将是我们今后探讨的课题。

表 2 不同参数取值情况下的实验结果

α	β	ρ	q_0	寻径成功率
1	2	0.200	0.7	0.5171170
1	2	0.100	0.7	0.6117220
1	2	0.001	0.7	0.5836360
1	2	0.500	0.7	0.4389800
1	4	0.100	0.7	0.6394930
4	1	0.100	0.7	0.1465350
2	4	0.100	0.7	0.6572440
1	4	0.100	0.5	0.5688541
2	4	0.100	0.9	0.6672570
2	4	0.100	0.3	0.5555560

由实验结果可知,ACA 能很好地找到网络最优解,并较好的收敛性和有效性。

4 结束语

本文采用蚁群算法求解 QoS 受限问题,它通过启发式方法取代传统数学方法解决 NP 完备问题,经过实验仿真,验证了该方法的有效性。

参考文献:

- [1] Gambardella LM, Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem[A]. Proceedings of ML-95, Twelfth International Conference on Machine Learning[C]. Morgan Kaufmann, 1995. 252-260.
- [2] Dorigo M, Maniezzo V, Colomi A. The Ant System: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1): 1-13.
- [3] Dorigo M, Gambardella LM. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [4] White T, Pagurek B, Oppacher F. ASGA: Improving the Ant System by Integration with Genetic Algorithms[A]. Proceedings of the 3rd Conference on Genetic Programming (GP/SGA'98)[C], 1998. 610-617.
- [5] Dorigo M, Caro GD. Ant Algorithms for Discrete Optimization[J]. Artificial Life, 1999, 5(3), 137-172.
- [6] Stutzle T, Hoos HH. MAX-MIN ant system[J]. Future Generation Computer System, 2000, 16(8): 889-914.
- [7] 张素兵, 吕国英, 刘泽民, 等. 基于蚂蚁算法的 QoS 路由调度方法[J]. 电路与系统学报, 2003, 5(1).
- [8] 王征应, 石冰心. 基于启发式遗传算法的 QoS 组播路由问题求解[J]. 计算机学报, 2001, 24(1).
- [9] 李生红, 刘泽民, 周正. ATM 网上基于蚂蚁算法的 VC 路由选择方法[J]. 通信学报, 2000, 21(1).
- [10] 何小燕, 费翔, 罗军舟, 等. Internet 中一种基于遗传算法的 QoS 路由选择策略[J]. 计算机学报, 2000, 23(11).