

一种新的混沌蚁群算法 及其在 QoS 组播路由优化问题中的应用

孔 笋, 陈增强

(南开大学 信息技术科学学院, 天津 300071)

摘 要: 基于 QoS 的组播路由问题是通过发现具有某种相关性能约束的最佳组播树, 来更好地利用网络资源以支持应用的 QoS 需求, 作为以 QoS 为中心的网络体系结构中不可缺少的组成部分, 目前已成为网络研究领域的重要内容和热点问题. 针对多约束条件下的 QoS 组播路由问题, 提出一种新的混沌蚁群算法. 该算法基于传统的蚁群算法所存在的不足, 利用混沌优化算法对蚁群算法的运行参数进行动态地优化选择, 自适应地改进了全局搜索能力和收敛性. 仿真实验结果表明, 混沌蚁群算法比该文提到的遗传算法及蚁群算法在解决多约束组播路由问题上具有更好的性能.

关键词: QoS; 组播路由; 混沌算法; 蚁群算法; 参数优化

中图分类号: TP183; TN949.291 **文献标志码:** A **文章编号:** 1673-4785(2010)06-0498-07

A new chaotic ant colony optimization algorithm and its application in a QoS multicast routing problem

KONG Sun, CHEN Zeng-qiang

(College of Information Technical Science, Nankai University, Tianjin 300071, China)

Abstract: QoS-based multicast routing can take advantage of network resources to support an application with QoS requirements by searching for the optimal multicast tree with some performance constraints. This problem, which is an indispensable part of QoS-centered network architecture, has become an important issue in network domain research. A new chaotic ant colony optimization algorithm was proposed for a multi-constrained QoS multicast routing problem. To overcome the deficiencies of traditional ant colony algorithms, this algorithm uses a chaotic optimization algorithm to dynamically select parameters of the ant colony algorithm and improves global searching and convergence abilities. Simulation results show that this chaotic ant colony optimization algorithm performs better than the genetic algorithm and ant colony optimization mentioned here for solving a QoS multicast routing problem with multiple constraints.

Keywords: QoS; multicast routing; chaos optimization algorithm; ant colony optimization algorithm; parameter optimization

QoS 组播路由的目的是在网络中寻找同时满足多用户对线路的带宽、延迟、延迟抖动、费用要求的路由, 即向多用户提供端到端的服务质量保证. Z. Wang 等学者已经证明了包含 2 个及以上的约束条件的 QoS 网络路由是一个 NP 完全问题^[1], 许多文献提出了解决该问题的有效算法. 目前基于智能优

化的组播路由算法受到了广泛研究, 应用较成熟的算法有遗传算法、神经网络算法、蚁群算法等^[2-6].

其中, 蚁群算法 (ant colony algorithm, ACA) 在深入研究中显示出了在求解复杂优化问题方面的优越性, 被广泛用于解决各种具有 NP 难的问题. 蚁群算法中的运行参数的选取对算法的收敛速度、全局、局部搜索能力有着至关重要的影响, 目前对于蚁群算法运行参数的选取通常都是针对具体问题通过大量的数字仿真实验确定, 缺乏理论的支持, 因此传统的蚁群算法很难取得较好的寻优性能. 在组播路由

收稿日期: 2009-11-24.

基金项目: 国家“863”计划资助项目 (2009AA04Z132); 国家自然科学基金资助项目 (60774088).

通信作者: 孔 笋. E-mail: ksuser@mail.nankai.edu.cn.

问题中,文献[7]提出一种用遗传蚁群算法来求解 QoS 组播路由问题,采用遗传算法对蚁群算法的 4 个控制参数进行编码、优化操作,为参数的选择提供了依据,更快地引导蚁群系统找到全局最优解;文献[8]基于蚁群算法的组播路由问题中,设计了一种自适应的挥发因子,用来控制算法的全局搜索能力;文献[9]提出一种改进的自适应蚁群算法,在信息素更新策略中引入全局最优系数,然后以全局最优系数为条件来自适应调整挥发因子和信息素强度常数。

考虑到混沌优化算法具有随机性、遍历性、规律性的搜索特点,一些研究者将混沌优化算法与蚁群算法相结合来提高蚁群算法性能^[10-11],主要有以下几种结合形式:1)采用混沌初始化进行改善个体质量;2)在调整信息量中,加入混沌扰动,以使解跳出局部极值区间;3)使用混沌搜索算子在当前迭代的全局最优解附近搜索更好的解。

与前面的混沌蚁群算法不同,文中结合蚁群算法的参数特性,提出了一种新的混沌蚁群优化算法,其基本思想是采用混沌搜索对蚁群算法中的控制参数进行优化,从而得到更好的参数组合使蚁群系统能更好、更快地找到全局最优解。文中将其应用于求解 QoS 组播路由问题,仿真结果显示混沌蚁群算法的求解性能要优于遗传算法及蚁群算法。

1 QoS 组播路由问题描述

网络模型表示为赋权图 $G = (V, E)$ ^[3], 式中 V 是图中所有网络节点组成的集合, E 是网络双向链路的集合, 每一条边表示 2 个节点间的通信路径, 假设网络是对称的。 $s \in V$ 为源点, $M \in \{V - \{s\}\}$ 为终点。

对于任一网络节点 $n \in V$, 定义 4 种属性, 分别为: 延迟函数 $\text{delay}(n)$ 、延迟抖动函数 $\text{delay_jitter}(n)$ 、包丢失率函数 $\text{packet_loss}(n)$ 、费用函数 $\text{cost}(n)$ 。

对于任意链路 $e \in E$, 定义 4 种属性: 延迟函数 $\text{delay}(e)$, 延迟抖动函数 $\text{delay_jitter}(e)$, 带宽函数 $\text{bandwidth}(e)$ 和费用函数 $\text{cost}(e)$ 。

对于给定的源点 $s \in V$, 终点集合 M , 节点 $t \in M$, s 和 M 组成的组播树 $T(s, M)$ 存在下列关系:

- 1) $\text{delay}(P_T(s, t)) = \sum_{e \in P_T(s, t)} \text{delay}(e) + \sum_{n \in P_T(s, t)} \text{delay}(n)$;
- 2) $\text{cost}(T(s, M)) = \sum_{e \in T(s, M)} \text{cost}(e) + \sum_{n \in P_T(s, M)} \text{cost}(n)$;

$$3) \text{bandwidth}(P_T(s, t)) = \min \{ \text{bandwidth}(e), e \in P_T(s, t) \};$$

$$4) \text{delay_jitter}(P_T(s, t)) = \sum_{e \in P_T(s, t)} \text{delay_jitter}(e) + \sum_{n \in P_T(s, t)} \text{delay_jitter}(n);$$

$$5) \text{packet_loss}(P_T(s, t)) = 1 - \prod_{n \in P_T(s, t)} (1 - \text{packet_loss}(n)).$$

式中: $P_T(s, t)$ 为组播树 $T(s, M)$ 上源点 s 到终点 t 的路由路径。以下给出 QoS 组播路由问题中约束条件的定义: 进行 QoS 路由的目的寻找一棵组播树 $T(s)$, 满足:

- 1) 延迟约束: $\text{delay}(P_T(s, t)) \leq D$;
- 2) 带宽约束: $\text{bandwidth}(P_T(s, t)) \geq B$;
- 3) 延迟抖动约束: $\text{delay_jitter}(P_T(s, t)) \leq DJ$;
- 4) 包丢失率约束: $\text{packet_loss}(P_T(s, t)) \leq PL$;
- 5) 费用约束: 在满足上述 4 个约束条件下,

$\text{cost}(T(s, M))$ 最小。

其中: B 、 D 、 DJ 、 PL 分别代表业务对网络带宽、延迟、延迟抖动、包丢失率的约束限制。在本模型中假设所有的组播终点的带宽约束、延迟、延迟抖动和包丢失率约束均相同。

2 蚁群算法及参数分析

本文将 Dorigo 等人提出的 ACS 算法^[12-13]中的选路方式引入到基本的蚂蚁算法中。下面针对 n 个城市的旅行商问题 (traveling salesman problem, TSP) 给出相应的算法模型。

1) 路径选取原则。

蚁群算法是一种基于模型的搜索算法, 它的搜索过程也是解的构造过程。对于求解 TSP 问题, 当蚂蚁 k 在当前城市 i 选择下一个将要移动的城市 s 时, 依据式 (1)、(2) 给出的伪随机比例规则进行。

$$s = \begin{cases} \arg \max_{j \in \text{allowed}_k} \{ \tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t) \}, & \text{if } q \leq q_0; \\ S, & \text{else.} \end{cases} \quad (1)$$

$$p_{ij}^k = \begin{cases} \tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t) / \sum \tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t), & j \in \text{allowed}_k; \\ 0, & \text{else.} \end{cases} \quad (2)$$

式中: $\text{allowed}_k = \{0, 1, \dots, n-1\}$ 表示蚂蚁 k 下一步允许选择的的城市集合; τ_{ij} 为边 (i, j) 上的信息素强度; η_{ij} 为边 (i, j) 的能见度; α 为信息启发式因子, 表示信息素对路径选择的重要性; β 为期望启发式因子, 表示启发信息对路径选择的重要性; q 是在 $[0, 1]$ 之间的随机数。

1] 间均匀分布的随机数; q_0 是一个可调参数 ($0 \leq q_0 \leq 1$); S 为根据是式(2)给出的概率分布所选出的一个随机变量。

2) 局部信息素更新规则。

当所有蚂蚁完成一次循环后, 各路径上的信息量要根据式(3)进行调整。

$$\tau_{ij}(t+n) = (1 - \rho_{\text{local}})\tau_{ij}(t) + \rho_{\text{local}}\Delta\tau_{ij},$$

$$\rho_{\text{local}} \in (0, 1). \quad (3)$$

式中: $\Delta\tau_{ij} = \sum_{k=1}^M \Delta\tau_{ij}^k$, ρ_{local} 表示路径上信息的蒸发系数; $1 - \rho_{\text{local}}$ 表示信息的保留系数; $\Delta\tau_{ij}$ 表示本次循环路径 ij 上信息的增量; $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在本次循环中留在路径 ij 上的信息量, 表示为

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & \text{若第 } k \text{ 只蚂蚁在本次循环中经过 } ij; \\ 0, & \text{其他.} \end{cases}$$

式中: Q 为常数, L_k 表示第 k 只蚂蚁在本次循环中所走过的路径的长度。

3) 全局信息素更新规则。

所有蚂蚁走完全程, 按式(4)进行信息素更新。

$$\tau_{ij}^{\text{new}} = (1 - \rho_{\text{global}})\tau_{ij}^{\text{old}} + \rho_{\text{global}}\Delta\tau_{ij},$$

$$\rho_{\text{global}} \in (0, 1). \quad (4)$$

式中: ρ_{global} 表示全局信息素的挥发系数; $\Delta\tau_{ij}$ 表示为:

$$\Delta\tau_{ij} = \begin{cases} 1/L_{\text{best}}, & \text{若 } ij \text{ 为全局最优路径所经过的边;} \\ 0, & \text{其他.} \end{cases}$$

式中: L_{best} 为全局最佳路径长度。

蚁群算法中的 5 个控制参数 q_0 、 α 、 β 、 ρ_{local} 、 ρ_{global} 的选取^[14-16]对算法的性能有较大的影响。式中可调参数 q_0 取值较大, 则多数蚂蚁易选择信息量最大的边, 在搜索过程中可能容易出现多数蚂蚁搜索到相同的路径, 使得搜索到的解空间较小, 不利于发现全局最优解, 算法容易收敛到局部最优解; 若 q_0 取值较小, 则信息量最大的边被选择的概率小, 其他边被选择的概率大, 能扩大搜索到的解空间, 但搜索呈现一定的盲目性, 不容易收敛。信息启发式因子 α 的大小反映了信息素因素作用的强度, 其值越大, 蚂蚁选择以前走过路径的可能性越大, 搜索的随机性减弱, 因此如果 α 值过大会使蚁群的搜索过早陷于局部最优, 而过小则会使算法收敛速度减慢。期望值启发因子 β 的大小反映了先验性、确定性因素作用的强度。其值越大, 蚂蚁在某个局部点上选择局部最短路径的可能性越大, 算法的随机性减弱, 易于陷入局部最优; 而 β 过小, 将导致蚂蚁群体陷入纯粹的随机搜索, 很难找到最优解。信息素挥发因子 ρ_{local} 、 ρ_{global} (这里统称 ρ) 的大小直接关系到蚁群算法的全局搜

索能力及其收敛速度。当 ρ 较大时, 由于信息正反馈的作用占主导地位, 以前搜索过的路径被再次选择的可能性过大, 搜索的随机性减弱; 反之, 当 ρ 很小时, 信息正反馈的作用相对较弱, 搜索的随机性增强, 因此蚁群算法收敛速度很慢。

3 基于混沌算法的蚁群算法参数优化设计

3.1 混沌优化

混沌优化^[17]是一种较新的优化算法, 它利用混沌序列的随机性、遍历性和初值敏感性来提高随机优化算法的效率。Logistic 序列是这类算法中常用的混沌序列。它可以用式(5)来描述。

$$x_{k+1} = \mu \cdot x_k(1.0 - x_k). \quad (5)$$

式中: μ 为常数, 取值在 $[3.56, 4.00]$ 之间。

不失一般性地, 假设待优化问题为如下 n 维函数的最小值问题:

$$\text{Min } f(x_1, x_2, \dots, x_n).$$

式中: $x_i \in [a_i, b_i]$, $i = 1, 2, \dots, n$ 。

其基本思想是首先产生一组与优化变量相同数目的混沌变量, 用类似载波的方式将混沌引入优化变量使其呈现混沌状态, 同时把混沌运动的遍历范围放大到优化变量的取值范围, 然后直接利用混沌变量搜索。由于混沌运动具有随机性、遍历性、对初始条件敏感性等特点, 基于混沌的搜索技术无疑会比其他随机搜索更具优越性。

3.2 参数优化算法设计

本文提出的参数优化算法的思想是将蚁群算法的运行参数作为混沌算法的优化对象, 在每一次迭代过程中, 使用混沌搜索的当前值来运行蚁群算法求解一标准优化问题, 并使用适应值评价函数对求解性能做出评价。

对于任一目的节点 $t_i \in M (i = 1, 2, \dots, P)$, 蚁群以节点为单位进行寻径, 并通过精简处理除去不满足带宽约束的路径, 构造混沌蚁群算法, 求解满足所有约束条件且总费用最小的组播路由。

基于上述规定和准则, 构造的混沌蚁群算法具体步骤如下:

1) 初始化参数。假定网络中有 N 个节点, 给定 M 只蚂蚁, 循环次数为 K 。给定 q_0 、 α 、 β 、 ρ_{local} 、 ρ_{global} 初始值及集合 Ω_i 中各备选路径的信息素初始值, 给出各节点 $(d, \text{DJ}, \text{PL}, c)$ 的值, 以及每条存在边 (d, DJ, b, c) 的值, 并给出约束条件的 D 、 DJ 、 PL 、 B 的值。

2) 蚂蚁从源节点开始按路径选择准则(1)随

机选择下一个要行走的节点,启发函数 η 取 $1/(\text{cost}(e_{ij}) + \text{cost}(n_j))$,期望值启发因子 β 采用 β/k 代替,使期望值在迭代的后期减少对路径选择的干扰,避免陷入局部最优,加快逼近最优解的速度.当每只蚂蚁对任一目的节点选择了一条路径,则对该路径进行局部信息素更新;当 M 只蚂蚁对所有的目的节点都选择了一条到达的路径后,根据目标函数^[3]计算各蚂蚁的所对应的目标函数值,并进行比较,获得当前迭代最优组播树和全局最优组播树,若当前迭代最优组播树包含环路,则逆序查询,直到找到无环路组播树,将该树作为当前迭代最优组播树,最后对全局最优组播路径进行信息素的更新.对于每项约束,本文假设各目的节点的约束限制相同,且 $\text{delay}(P_T(s, t))$ 、 $\text{delay_jitter}(P_T(s, t))$ 、 $\text{packet_loss}(P_T(s, t))$ 分别取组播树中源节点到达各目的节点的最大约束值,目标函数为式(6)

$$f_T = \frac{1}{\text{cost}[T(s, M)]}(f_d + f_{dj} + f_{pl}),$$

$$f_d = \Phi_d(\text{delay}(P_T(s, t)) - D),$$

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

$$f_{dj} = \Phi_{dj}(\text{delay_jitter}(P_T(s, t)) - DJ),$$

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

$$f_{pl} = \Phi_{pl}(\text{packet_loss}(P_T(s, t)) - PL),$$

$$(\Phi_{pl}(z) = \begin{cases} 1, & z \leq 0 \\ r_{pl}, & z > 0 \end{cases}). \quad (6)$$

式中: $\Phi_d(z)$ 是延时度量的惩罚函数,当满足延时约束 ($\text{delay}(P_T(s, t)) \leq D$) 时,其值为 1,否则等于 r_d ($0 \leq r_d \leq 1$); $\Phi_{dj}(z)$ 是延时抖动度量的惩罚函数,当满足延时抖动约束 ($\text{delay_jitter}(P_T(s, t)) \leq DJ$) 时,其值为 1,否则等于 r_{dj} ($0 \leq r_{dj} \leq 1$); $\Phi_{pl}(z)$ 是包丢失率度量的惩罚函数,当满足包丢失率约束 ($\text{packet_loss}(P_T(s, t)) \leq PL$) 时,其值为 1,否则等于 r_{pl} ($0 \leq r_{pl} \leq 1$); r_d 、 r_{dj} 、 r_{pl} 三者的值的大小决定惩罚的程度.在本文试验仿真中, $r_d = r_{dj} = r_{pl} = 0.5$.

3) 将 q_0 、 α 、 β 、 ρ_{local} 、 ρ_{global} 作为待优化的混沌变量,并给定各参数相应的取值范围,利用 Logistic 映射进行混沌搜索.

① 首先将 5 个参数变量利用式(7)映射为混沌变量,取值范围为 $[0, 1]$,

$$x_i^k = (y_i^k - a_i) / (b_i - a_i). \quad (7)$$

式中: y_i^k 为优化变量, $y_i \in [a_i, b_i]$, $i = 1, 2, \dots, n$, x_i^k 为混沌变量.

② 用式(5)对混沌变量进行优化搜索.

③ 将混沌变量映射为优化变量,如式(8):

$$y_i^k = a_i + x_i^k(b_i - a_i). \quad (8)$$

④ 利用适应值函数对当前所得到参数运行后的算法性能作出评价,适应值评价函数设计如下:

$$\text{Fit}_k(s) = \delta_1 L_k(s) + \delta_2 V_k(s) + \delta_3 G_k(s),$$

$$L_k(s) = L_{\text{ave}},$$

$$V_k(s) = e^{-v/K},$$

$$G_k(s) = 1/G_{\text{sum}}.$$

式中: $\text{Fit}_k(s)$ 为第 k 次搜索所找到的算法参数所对应的适应值; δ_1 、 δ_2 、 δ_3 为权系数,并满足 $\delta_1 + \delta_2 + \delta_3 = 1$; $L_k(s)$ 表示蚁群算法搜索到的最优路径的能力; L_{ave} 为 M 只蚂蚁寻找到的组播树的平均目标函数值; $V_k(s)$ 表示蚁群算法的收敛速度; v 是蚁群当前迭代次数; K 为总迭代次数; $G_k(s)$ 表示蚁群算法的全局搜索能力; G_{sum} 为蚁群 5 个参数变量的和.

⑤ 经比较如果当前搜索到的参数使得适应度函数偏大,则采用该参数进行下一轮寻径,若适应度函数偏小,则继续混沌搜索,直到寻找到更优的参数或达到了最大步数.

4) 重复执行以上步骤,直到寻找到最优组播树.

4 仿真实验及比较

本文通过程序实现了混沌蚁群算法在解决 QoS 组播路由优化的应用,仿真在 Matlab 7.0 环境下实现.为了验证本文算法的有效性,本文采用了文献[3]的网络结构模型进行实验仿真,并将混沌蚁群算法与应用于组播路由问题的遗传算法^[3]、蚁群算法^[8]进行比较,网络拓扑图如图 1 所示:

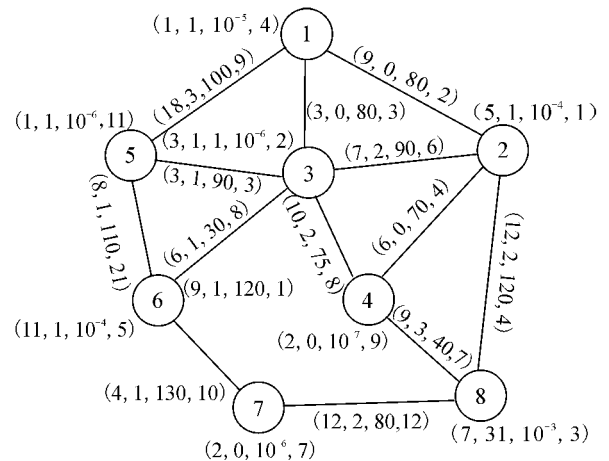


图 1 8 节点网络模型^[3]

Fig. 1 Eight-node network^[3]

仿真实验以节点 1 为源节点,以节点 2、节点 4、节点 5、节点 7 为目的节点,寻找满足约束条件的最优路径.参数 q_0 、 α 、 β 、 ρ_{local} 、 ρ_{global} 为优化对象,初始值

分别为0.12、4、1.1、0.13、0.22,其参数取值范围如表1所示.实验中,蚂蚁数 $M=30$,最大迭代次数 $K=20$,最大混沌迭代次 $\text{Max}C=10$, δ_1 、 δ_2 、 δ_3 分别为0.8、0.1、0.1.

表1 蚁群算法参数取值范围

Table 1 Parameters range of ant colony algorithm

运行参数	参数允许范围	运行参数	参数允许范围
q_0	(0,1)	β	(1,3)
α	(3,7)	$\rho_{\text{local,global}}$	(0,1)

为了更好地对比,本文同文献[3]一样,对于同一网络模型,使用了2组约束条件 $B=70, D=46, DJ=8, PL=0.001$ 和 $B=70, D=50, DJ=6, PL=0.001$ 分别进行仿真,经过求解得到的最佳组播树如图2.

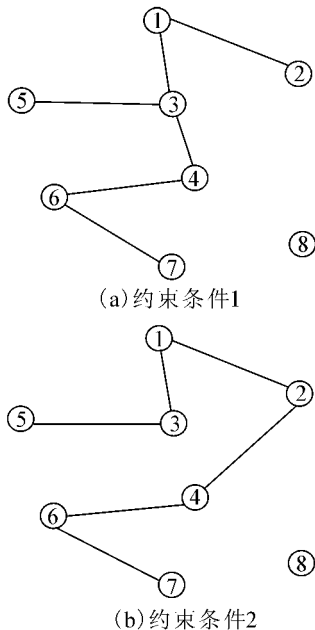


图2 最优组播树

Fig.2 Optimal multicast tree

从文献[3]采用的启发式遗传算法仿真结果图中,可以看出遗传算法经过12次迭代开始趋近于最优解.在相同网络条件下的蚁群算法如图3,图5经过10次迭代开始收敛于最优解,且蚁群算法需要较多的蚂蚁数,否则易陷入局部最优,这里蚁群算法中的蚂蚁数取为50.图4、图6为所采用的混沌蚁群算法得到的仿真结果,与遗传算法和蚁群算法相比,基于混沌优化的蚁群算法有较强的自适应性,在适应值函数的指导下,加快了收敛速度,另一方面引入的 $1/(q_0 + \alpha + \beta + \rho_{\text{local}} + \rho_{\text{global}})$ 保证了算法的全局搜索能力,算法的代价、延时、延时抖动曲线从第4代开始收敛,且具有较好的稳定性,在算法运行的后期每一代的操作都能得到最佳组播树,这说明了混沌蚁

群算法有更好的收敛性.

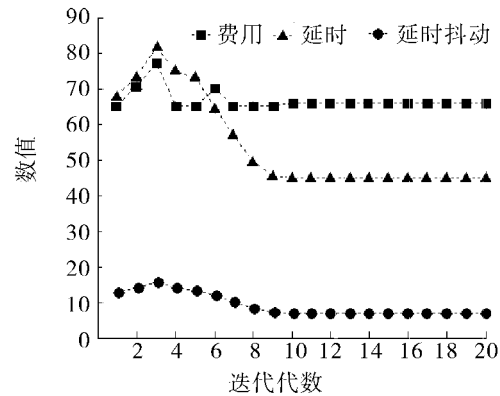


图3 $B=70, D=46, DJ=8, PL=0.001$ 时蚁群算法组播树代价、延时和延时抖动随迭代代数变化曲线

Fig.3 Curves of cost, delay and delay jitter of multicast tree of ACS when $B=70, D=46, DJ=8$, and $PL=0.001$

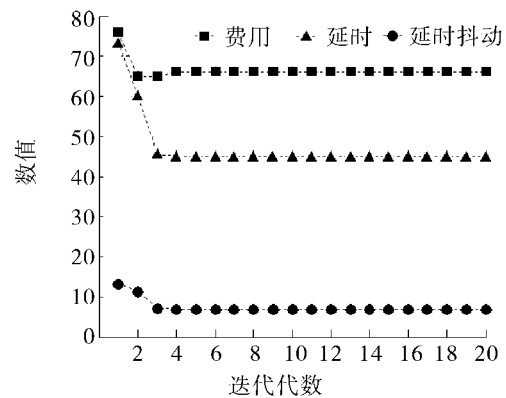


图4 $B=70, D=46, DJ=8, PL=0.001$ 时混沌蚁群算法组播树代价、延时和延时抖动随迭代代数变化曲线

Fig.4 Curves of cost, delay and delay jitter of multicast tree of chaos ant colony optimization algorithm when $B=70, D=46, DJ=8$, and $PL=0.001$

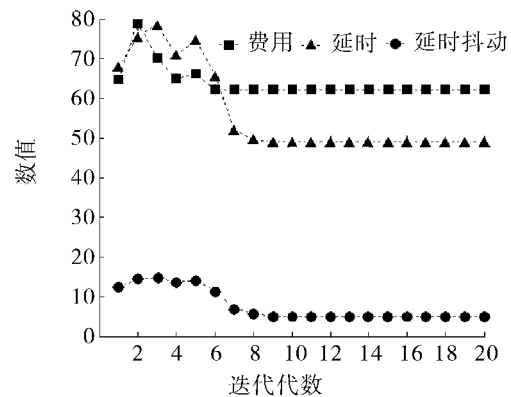


图5 $B=70, D=50, DJ=6, PL=0.001$ 时蚁群算法组播树代价、延时和延时抖动随迭代代数变化曲线

Fig.5 Curves of cost, delay and delay jitter of multicast tree of ACS when $B=70, D=50, DJ=6$, and $PL=0.001$

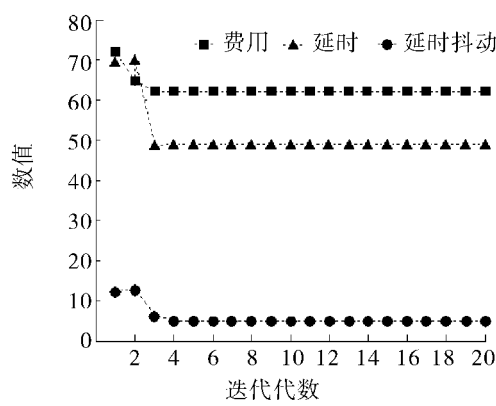


图6 $B=70,D=50,DJ=6,PL=0.001$ 时混沌蚁群算法组播树代价、延时和延时抖动随迭代代数变化曲线

Fig.6 Curves of cost, delay and delay jitter of multicast tree of chaos ant colony optimization algorithm when $B=70,D=50,DJ=6$, and $PL=0.001$

本文引用了2组约束条件,2个约束条件体现了混沌蚁群算法在不同的环境下,寻优性能有一定差别。如在启发函数 η 取为当前路径代价的倒数的指导下,第1组约束条件要花费较长的时间才能寻找到最优解,易于陷入局部最优;基于相同的启发函数,第2组约束很快能够找到最优解。虽然2个仿真结果有所差异,但与遗传、蚁群算法相比,最终都以较快的速度收敛于最优解。表2给出了在混沌搜索下参数随迭代次数变化的情况。结合前面的蚁群参数对算法的性能影响分析,从表中可以看出,随迭代次数的增长,算法由初期较强的全局搜索能力发展到后期较强的局部搜索能力,实现了参数自适应。

表2 优化参数

Table 2 Optimized parameters

蚁群参数	第1组约束条件 $B=70,D=46,DJ=8,PL=0.001$			
	第2组约束条件 $B=70,D=50,DJ=6,PL=0.001$			
	1	3	10	20
q_0	0.12	0.422 4	0.975 9	0.975 9
	0.12	0.975 9	0.975 9	0.975 9
α	4.1	5.583 9	5.583 9	5.583 9
	4.1	5.583 9	5.583 9	5.583 9
β	1.1	1.404 8	2.231 2	2.231 2
	1.1	2.231 2	2.231 2	2.231 2
ρ_{local}	0.13	0.452 4	0.990 9	0.990 9
	0.13	0.990 9	0.990 9	0.990 9
ρ_{global}	0.22	0.998 2	0.861 0	0.861 0
	0.22	0.861 0	0.861 0	0.861 0

另外,为了验证算法在大规模网络下的有效性,采用了随机化的方法^[18]生成一些具有实际网络特性的拓扑模型:首先随机产生一系列节点,按照一定的概率生成机制建立链路的连接,节点的平均度为4;随机给定图中节点和各边的属性值(参照上述8节点的网络属

性值)。蚁群大小、运行代数及混沌搜索次数随网络节点变化的情况如表3所示。

表3 蚁群大小、运行代数及混沌搜索次数随网络节点变化的情况

Table 3 Colony size, running algebra and chaos searches algebra with the changes of the network nodes

网络节点	链路数	蚁群大小	蚁群迭代次数	混沌搜索次数
N20	E78	20	30	10
N40	E167	100	50	20
N60	E256	130	60	30
N80	E342	160	70	40
N100	E439	180	90	50
N120	E514	200	100	60

下面给出了提出的混沌蚁群算法与文献[8]中的蚁群算法(WACO)的性能对比曲线。从图7中可以看出新的混沌蚁群算法在大规模的网络中所得到的最优组播树代价均比文献[8]中的蚁群算法所求得的最优组播树代价要低。另外,图8显示出了混沌蚁群算法具有较快的收敛速度,实验表明该算法可适应于大规模网络,并可获得比文献[8]更好的寻优结果。

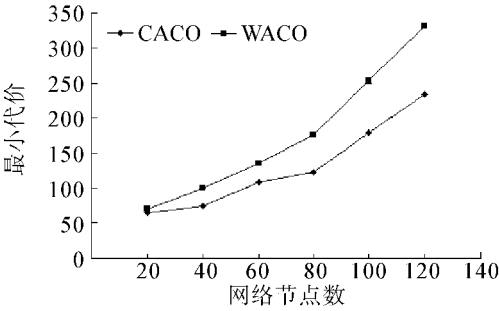


图7 最小代价对比曲线

Fig.7 Comparison of least-cost performance

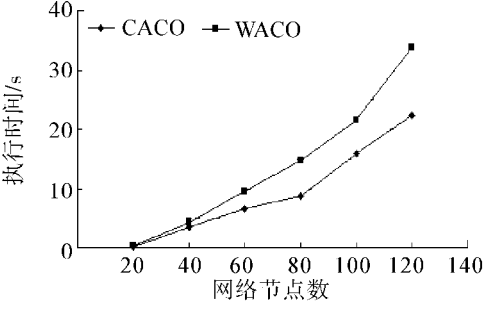


图8 收敛速度对比曲线

Fig.8 Comparison of convergence speed

5 结束语

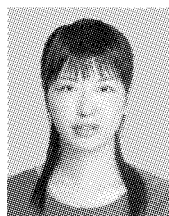
提出了一种基于混沌搜索的蚁群参数优化算法,首次利用混沌搜索来自适应地调整蚁群的运行参数,并以平均代价大小、运行时间及全局搜索能力为性能指标函数来评价当前参数对算法的影响,最后应用该算法来求解包含延时、延时抖动、带宽、丢

包率等约束条件,并保证费用最小的 QoS 组播路由问题. 该算法不仅能准确地找到最优组播树,还在原有算法的基础上有效地加快了收敛速度,提高了算法寻找最优解的效率. 实验结果证明了新算法的有效性,并具有一定的推广性.

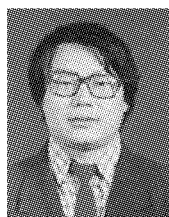
参考文献:

- [1] WANG Z, CROWCROFT J. Quality of service for supporting multimedia applications[J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228-1234.
- [2] FEI X, LUO J Z, WU J Y, GU Q Q. QoS Routing based on genetic algorithm[J]. Computer Communications, 1999, 22(9): 1394-1399.
- [3] 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): 55-61.
WANG Zhengying, SHI Bingxin. Solving QoS multicast routing problem based on heuristic genetic algorithm[J]. Chinese Journal of Computers, 2001, 24(1): 55-61.
- [4] 孙文生, 刘泽民. 组播路由调度的神经网络方法[J]. 通信学报, 1998, 19(11): 1-6.
SUN Wensheng, LIU Zemin. Multicast routing based neural networks[J]. Journal on Communications, 1998, 19(11): 1-6.
- [5] ZHANG Li, CAI Lianbo, LI Meng, WANG Fahui. A method for least-cost QoS multicast routing based on genetic simulated annealing algorithm[J]. Computer Communications, 2009, 32: 105-110.
- [6] 李生红, 潘理, 诸鸿文, 刘泽民. 基于蚂蚁算法的组播路由调度方法[J]. 计算机工程, 2001, 27(4): 63-65.
LI Shenghong, PAN Li, ZHU Hongwen, LIU Zemin. Ant-algorithm based multicast routing[J]. Computer Engineering, 2001, 27(4): 63-65.
- [7] 孙力娟, 王汝传. 基于蚁群算法和遗传算法融合的 QoS 组播路由求解[J]. 电子学报, 2006, 34(8): 1391-1395.
SUN Lijuan, WANG Ruchuan. Solving QoS multicast routing problem based on the combination of ant colony algorithm and genetic algorithm[J]. Acta Electronica Sinica, 2006, 34(8): 1391-1395.
- [8] WANG Y, XIE J. Ant colony optimization for multicast routing[C]//The 2000 IEEE Asia-Pacific Conference on Circuits and Systems. [S. l.], 2000: 54-57.
- [9] 陈杰, 张洪伟. 基于自适应蚁群算法的 QoS 组播路由算法[J]. 计算机工程, 2008, 34(13): 200-203.
CHEN Jie, ZHANG Hongwei. QoS multicast routing algorithm based on adaptive ant colony algorithm[J]. Computer Engineering, 2008, 34(13): 200-203.
- [10] 陈烨. 变尺度混沌蚁群优化算法[J]. 计算机工程与应用, 2007, 43(3): 68-70.
CHEN Ye. Scaleable chaotic ant colony optimization[J]. Computer Engineering and Applications, 2007, 43(3): 68-70.
- [11] 高尚. 解旅行商问题的混沌蚁群算法[J]. 系统工程理论与实践, 2005, 25(9): 100-104.
- GAO Shang. Solving traveling salesman problem by chaos ant colony optimization algorithm[J]. System Engineering-Theory & Practice, 2005, 25(9): 100-104.
- [12] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating Agent[J]. IEEE Transactions on Systems, Man and Cybernetics, 1996, 26(1): 29-41.
- [13] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 41(1): 53-66.
- [14] 叶志伟, 郑肇葆. 蚁群算法中参数设置的研究[J]. 武汉大学学报: 信息科学版, 2004, 29(7): 597-601.
YE Zhiwei, ZHENG Zhaobao. The research on the parameter in ant colony algorithm[J]. Geomatics and Information Science of Wuhan University, 2004, 29(7): 597-601.
- [15] 吴春明, 陈治, 姜明. 蚁群算法中系统初始化及系统参数的研究[J]. 电子学报, 2006, 34(8): 1530-1532.
WU Chunming, CHEN Zhi, JIANG Ming. The research on initialization of ants system and configuration of parameters for different TSP problems in ant algorithm[J]. Acta Electronica Sinica, 2006, 34(8): 1530-1532.
- [16] 朱庆保. 蚁群优化算法的收敛性分析[J]. 控制与决策, 2006, 21(7): 763-770.
ZHU Qingbao. Analysis of convergence of ant colony optimization algorithms[J]. Control and Decision, 2006, 21(7): 763-770.
- [17] 李兵, 蒋慰孙. 混沌优化方法及其应用[J]. 控制理论及应用, 1997, 14(4): 613-615.
LI Bing, JIANG Weisun. Chaos optimization method and its application[J]. Control Theory & Applications, 1997, 14(4): 613-615.
- [18] WAXMAN B M. Routing of multipoint connections[J]. IEEE Journal on Selected Areas in Communications, 1988, 6(9): 1617-1622.

作者简介:



孔 笋,女,1982 年生,博士研究生,主要研究方向为智能优化与鲁棒控制.



陈增强,男,1964 年生,教授,博士生导师,自动化系主任. 主要研究方向为智能预测控制、混沌系统与复杂动态网络、多智能体系统控制. 发表学术论文 100 多篇,其中在 IEEE 刊物上发表 5 篇(包括长文 1 篇),被 SCI 和 EI 检索 100 余篇.