

文章编号 :1001-2486(2007)02-0117-06

基于树分解/合并策略的 QoS 多播路由方法*

陈 岩 杨华江 朱华勇 沈林成

(国防科技大学 机电工程与自动化学院 湖南 长沙 410073)

摘 要 :针对多约束 QoS 多播路由的 NP-Complete 特性 ,提出一种可控的多播树分解与合并策略 ,使多播树的生成在兼顾低费用的同时具有多样性 ,有效克服多播路由优化的局部极值问题。基于该策略设计蚁群算法 ,分解蚂蚁种群为与多播目标点相对应的蚂蚁子群 ,引入基于“死点”惩罚和多播树奖惩的信息素更新机制 ,提高了算法的收敛速度。仿真实验表明 ,该方法能有效地解决 QoS 多播路由问题 ,且随着网络规模的增大保持了良好的性能。

关键词 :多播路由 ;多播树 ;蚁群算法

中图分类号 :TP393.04 文献标识码 :A

A Decomposition/Composition-of-Tree Strategy-based Method for QoS Multicast Routing Problem

CHEN Yan ,YANG Hua-jiang ,ZHU Hua-yong ,SHEN Lin-cheng

(College of Mechatronics Engineering and Automation ,National Univ. of Defense Technology ,Changsha 410073 ,China)

Abstract :In view of the fact that the multiple constrained QoS multicast routing is a NP-Complete problem , a controllable strategy for decomposing and composing multicast tree is introduced. Different trees of little cost will be gained under the strategy , so that the problem of local best can be solved effectively. An ant colony algorithm , which divides the ant colony into ant assembly according to the ends of multicast , is designed on the basis of the strategy. The pheromone updating mechanism is improved with the “dead-point” punishment and the encouragement and punishment to the multicast tree strategy , so that the algorithm can converge fast. Simulation results demonstrate that the method can solve the QoS multicast routing problem effectively , and perform well as the size of network increases.

Key words :multicast routing ; multicast tree ; ant colony algorithm

多播服务是下一代战场数据分发系统的重要特征 ,而 QoS 多播路由是其必须解决的关键技术问题。研究表明 ,基于多个不相关可加性度量的 QoS 多播路由问题是 NP 完全问题^[1]。目前解决该类问题的主要方法为启发式算法^[2-5]。

蚁群算法是一种模拟蚂蚁觅食行为的仿生优化算法 ,具有并行搜索、群体寻优的特点 ,对于多播问题具有较强的可扩展性^[2]。对于 QoS 多播路由 ,文献 [3] 提出一种混合蚁群算法 ,将遗传算法与蚁群算法相合并 ,但是由于未采取有效措施抑制遗传算法的早熟现象 ,算法易陷入局部极值。文献 [4] 提出一种启发式蚁群算法 ,算法要求每只蚂蚁在一次循环中遍历所有目标点 ,由于文中采取对蚂蚁刚访问链路信息素衰减的局部更新策略 ,使算法有深度搜索的趋势 ,会造成大量的无效搜索 ,随着网络的增大将极大地降低算法效率。

为了能够快速、优质地实现 QoS 多播路由 ,本文在对 QoS 多播路由模型进行分析的基础之上 ,根据多播路由中约束条件下对各目标点“单播”路由相对独立的特性 ,首先将多播路由分解为对各目标点的单播路由 ,然后在带费用启发信息控制策略下 ,由“单播”路径再合并生成多播树 ,避免了算法早熟 ,提高了算法的全局搜索能力 ,同时在求解蚁群算法中引入了局部信息素惩罚更新机制和基于多播树奖惩

* 收稿日期 2006-12-27

基金项目 :国家自然科学基金重点资助项目(60234030)

作者简介 :陈岩(1962—)男 ,博士生。

的全局信息素更新机制,能有效缩小算法的搜索范围,提高算法收敛速度。

1 QoS 多播路由模型

设 N, P, E 表示网络, P 为网络节点集, E 为网络链路集, $E = \{(j, r) | j, r \in P \wedge j, r \text{ 为节点编号}\}$, $|P|$ 和 $|E|$ 分别表示网络节点数和链路数; $s \in P$ 为多播源点, $D \subseteq \{P - \{s\}\}$ 为多播目标节点集, $|D|$ 表示目标节点数。设 R^+ 表示正实数集, $\forall p \in P$ 有关于 QoS 指标的度量: 时延函数 $PD\alpha(p) : P \rightarrow R^+$, 时延抖动函数 $PDJ(p) : P \rightarrow R^+$, 丢报率函数 $PPI(p) : P \rightarrow R^+$, 费用函数 $PC\alpha(p) : P \rightarrow R^+$; 同理 $\forall e \in E$ 有: 时延函数 $ED\alpha(e)$, 时延抖动函数 $EDJ(e)$, 带宽函数 $EBW(e)$, 费用函数 $EC\alpha(e)$ 。

设有多播树 $T(s, D)$, $\mathcal{K}(s, d_i)$ 表示 T 中从 s 到 $d_i \in D$ 的路径, 则有如下关系:

1) 路径时延

$$D\alpha(\mathcal{K}(s, d_i)) = \sum_{e \in \mathcal{K}(s, d_i)} ED\alpha(e) + \sum_{p \in \mathcal{K}(s, d_i)} PD\alpha(p) \quad (1)$$

2) 路径时延抖动

$$DJ\alpha(\mathcal{K}(s, d_i)) = \sum_{e \in \mathcal{K}(s, d_i)} EDJ(e) + \sum_{p \in \mathcal{K}(s, d_i)} PDJ(p) \quad (2)$$

3) 路径瓶颈带宽

$$BW(l(\mathcal{K}(s, d_i))) = \min_{e \in \mathcal{K}(s, d_i)} EBW(e) \quad (3)$$

4) 路径丢报率

$$PI(\mathcal{K}(s, d_i)) = 1 - \prod_{p \in \mathcal{K}(s, d_i)} PPI(p) \quad (4)$$

5) 多播树费用

$$C\alpha(T(s, D)) = \sum_{e \in T(s, D)} EC\alpha(e) + \sum_{p \in T(s, D)} PC\alpha(p) \quad (5)$$

QoS 多播路由问题实际上是要在 N 中寻找一棵满足式(6)的多播树 $T(s, D)$:

$$\begin{aligned} & \min C\alpha(T(s, D)) \\ \text{s. t. } & \begin{cases} D\alpha(\mathcal{K}(s, d_i)) \leq D_{d_i} \\ DJ\alpha(\mathcal{K}(s, d_i)) \leq DJ_{d_i} \\ BW(\mathcal{K}(s, d_i)) \geq BW_{d_i} \\ PI(\mathcal{K}(s, d_i)) \leq PL_{d_i} \end{cases}, \quad i = 1, 2, \dots, |D| \end{aligned} \quad (6)$$

其中 $D_{d_i}, DJ_{d_i}, BW_{d_i}$ 和 PL_{d_i} 分别表示对应于目标点 d_i 的时延, 时延抖动, 带宽和丢报率路由约束; 为了便于讨论, 本文设定 D 中所有目标点各项路由约束相同, 分别为 Dd, DJ, BW, PL 。

2 约束条件下的多播树分解与合并

2.1 多播树分解

QoS 多播路由在仅考虑式(6)约束条件下, 对 D 中各目标点 $d_i, i = 1, 2, \dots, |D|$ 的传输路径路由具有一定的独立性, 即在不以多播树费用最小作为优化目标, 而仅考虑多播树的 QoS 指标约束时, 对 D 中任意目标点的“单播”路径路由, 不会影响其它目标点单播路径对约束条件的满足。因此, 多播树分解即是将对 D 的多播路由分解为对 D 中各目标点的单播路由。

2.2 多播树合并

通过多播树分解可以得到满足式(6)约束条件的单播路径, 将对应于 $d_i, i = 1, 2, \dots, |D|$ 的单播路径再合并即可实现对 D 满足式(6)约束条件的多播路由 $T^*(s, D)$ 。

2.2.1 问题说明与定义

由于对 $d_i, i = 1, 2, \dots, |D|$ 的单播路由独立, 因此在多播树合并时不能保证 $T^*(s, D)$ 是一棵树, T^*

(s, D)中可能包含环路。为此本文作如下定义：

定义1 链路支数 $Z(e) \quad \mathcal{K}(s, d_i) \quad i=1, 2, \dots, |D|$ 中经过链路 e 的数目为链路支数 $Z(e)$ 。

定义2 链路正支数 $Z^+(e)$ 设 $e=(j, r)$ 且 $j < r$ 称从 j 进入链路 e 的路径数目为 $Z^+(e)$; 否则称为负支数 $Z^-(e)$ 。

定义3 多播树干 链路 e 被称为多播树干当且仅当至少存在两条路径 $\mathcal{K}(s, d_i), \mathcal{K}(s, d_j)$ 满足：
(1) $e \in \mathcal{K}(s, d_i)$ 且 $e \in \mathcal{K}(s, d_j)$ (2) $\mathcal{K}(s, d_i)$ 和 $\mathcal{K}(s, d_j)$ 中 e 之前的链路按顺序相同。称满足上述条件的路径数为树干 e 权数 $R(e)$ (记非树干链路树干权数 $R(e)=1$)。

推论1 如果链路 $e \in \mathcal{K}(s, d_i)$ 为树干, 则 $\mathcal{K}(s, d_i)$ 中 e 之前的链路均为树干。

定义4 扩展多播树 $e \in T^*(s, D)$ 如果 $Z(e) \neq R(e)$ 则在网络 N 中加入多条与 e 具有相同链路、端点属性及与其它链路连接关系的扩展链路 e^* 与 e 构成扩展链路集 E^* 且使 $\forall e^* \in E^*$ 满足 (1) 如果 e 是树干, 则 $Z(e^*)=R(e)$ (2) 如果 e 不是树干, 则 $Z(e^*)=1$ 。称加入扩展链路后的多播树为扩展多播树, 如图1所示。

扩展多播树没有改变网络的拓扑结构, 只是便于从树的角度讨论问题; 下文中关于 L_i 合并多播树的讨论无特殊说明均指扩展多播树。

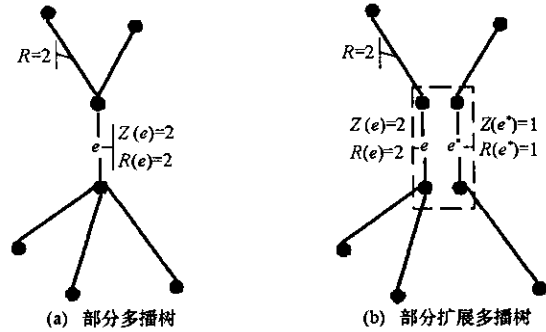


图1 扩展多播树

Fig. 1 Extended multicast tree

2.2.2 多播树合并策略

多播树的合并生成要兼顾多样性与费用优化, 为此

本文设计了包含路径集过滤和多播树概率生成的多播树合并策略：

1) 路径集过滤

设对 d_i 的单播路径集为 L_i , 路径集过滤即是去除 L_i 中的相同路径。设经过过滤后的路径集为 L_i^* , $i=1, 2, \dots, |D|$ L_i^* 中元素个数为 $|L_i^*|$ 。

2) 多播树概率生成

设一次多播树生成要合并 N 棵多播树, 每棵树的生成历经 $|D|$ 步, 每步即是按照合并概率从 L_i^* 中选择一条路径。设当前处于合并第 n 棵树的第 ζ 步, $1 \leq n \leq N, 1 \leq \zeta \leq |D|$, 前 $\zeta - 1$ 步合并得到不完全多播树 $\tilde{T}^*(s, D)$; 对 L_ζ^* 中路径定义如下参数：

定义5 路径合并度 $\chi(l)$ 设路径 l 与 $\tilde{T}^*(s, D)$ 组成不完全多播树 $(l + \tilde{T}^*(s, D))$; 如果在前 $n - 1$ 棵多播树中存在 v 棵, 这 v 棵的前 ζ 步不完全多播树与 $(l + \tilde{T}^*(s, D))$ 一致, 则 $\chi(l) = v$ 。

定义6 路径可行合并率 $\alpha(l) \quad \forall \bar{l} \in L_\zeta^*$, 记 $\chi^*(\bar{l}) = \begin{cases} \chi(\bar{l}) + 1 & \bar{l} = l \\ \chi(\bar{l}), & \text{others} \end{cases}$, 则 $\alpha(l) =$

$$\frac{1}{|L_\zeta^*|} \sum_{\bar{l} \in L_\zeta^*} (\chi^*(\bar{l}) - \frac{\sum_{\bar{l} \in L_\zeta^*} \chi^*(\bar{l})}{|L_\zeta^*|})^2$$

$\alpha(l)$ 反映了路径 l 可能配队成与前 $n - 1$ 棵树相重合的可能性, $\alpha(l)$ 越小, 则重合的概率越小。

定义7 路径相对可行合并率 $r_\alpha(l) \quad r_\alpha(l) = 1 - \frac{\alpha(l)}{\sum_{\zeta \in L_\zeta^*} \alpha(\zeta)}$

$r_\alpha(l)$ 描述了 L_ζ^* 中各路径生成多播树重合的相对概率, $r_\alpha(l)$ 越大, 则重合概率越小。

则在第 ζ 步配队时 L_ζ^* 中路径 l 的选择概率为：

$$p_l = \frac{(r_\alpha(l))^{l_1} \cdot \overline{C\alpha(l)^2}}{\sum_{\tau \in L_\zeta^*} (r_\alpha(\tau))^{l_1} \cdot \overline{C\alpha(\tau)^2}} \quad (7)$$

式中 $\overline{EC\alpha(l)} = 1 - \frac{EC\alpha(l)}{\sum_{l \in L_i^*} EC\alpha(l)}$, 反映了路径费用的启发信息; λ_1, λ_2 为权重因子, 反映了在合并生成多播树时, 树的多样性与费用启发信息的相对重要性。

3 QoS 多播路由蚁群算法

基于上述多播树分解—合并方法, 针对目标点 $d_i, i=1, 2, \dots, |D|$ 设置蚂蚁子群 $ant_i, i=1, 2, \dots, |D|$, 各子群在式(6)约束条件下依据状态转移规则独立进行单播路由, 子群间共享信息素 ρ_{ant_i} , 每次迭代结束后将得到满足约束条件的路径集 $L_i, i=1, 2, \dots, |D|$ 中各取一条路径进行合并, 可得多播树 $T^*(s, D)$ 。以最小化 $T^*(s, D)$ 代价为目标调整信息素, 引导后代 ant_i 搜索更有利于最小化多播代价的路径, 最终实现对式(6)的求解。

3.1 蚁群状态转移规则

蚂蚁 k 在单播路径搜索时, 根据下式由当前节点 j 选择后继节点 r :

$$r = \begin{cases} \arg \max_{u \in allowed_k(t)} \{ [\tau(j, \mu)]^\alpha \cdot [\eta(j, \mu)]^\beta \}, & q \leq q_0 \\ R, & \text{others} \end{cases} \quad (8)$$

其中 q 为 $[0, 1]$ 均匀分布的随机数, q_0 为算法参数, R 为由下式概率分布选出的随机变量。

$$p_{jr}(t) = \begin{cases} \frac{\tau_{jr}^\alpha(t) \cdot \eta_{jr}^\beta(t)}{\sum_{u \in allowed_k(t)} (\tau_{ju}^\alpha(t) \cdot \eta_{ju}^\beta(t))}, & r \in allowed_k \\ 0, & \text{others} \end{cases} \quad (9)$$

式中 $\tau_{jr}(t)$ 指 t 时刻链路 (j, r) 的信息素浓度, $\alpha \geq 0$ 为信息素启发因子, 表示轨迹的相对重要性, $\beta \geq 0$ 为期望启发因子, 表示能见度的相对重要性; $allowed_k(t)$ 表示 t 时刻蚂蚁 k 可以选择的后继节点集, $allowed_k(t) = C - Tabu_k(t)$, C 为与节点 j 存在链路且与之前搜索路径共同满足式(6)约束的节点集, $Tabu_k(t)$ 表示当前循环到 t 为止蚂蚁 k 所走过的节点集; $\eta_{ju}(t) = 1 / (EC\alpha(j, \mu) + PC\alpha(u))$ 为启发函数。

3.2 局部信息素更新规则

蚁群在进行单播路由时, 由于 QoS 指标约束以及禁忌表搜索限制, 会出现部分蚂蚁路由失败的情况, 该类蚂蚁携带有与路由相反的信息—其走过的部分链路不利于路由, 合理利用该类信息有助于避免不必要的搜索, 提高算法效率。

定义 8 死点 如果蚂蚁 k 位于节点 r , 且根据式(8)无后继节点可以选择, 则称节点 r 是蚂蚁 k 当前循环的死点。

蚂蚁如果走到死点那么当前搜索必定失败, 本文对引导蚂蚁走向死点的链路进行信息素惩罚, 减少该链路再次被搜索的可能性:

设蚂蚁 k 当前位于节点 r , 且为当前循环的死点, 前一时刻蚂蚁位于节点 j , 则对链路 (j, r) 信息素进行如下更新:

$$\tau_{jr}(t+1) = (1 - \mu)\tau_{jr}(t) + \mu(1 - \eta_{jr}(t))Q_1 \quad (10)$$

其中 $\tau_{jr}(t)$ 表示 t 时刻链路 (j, r) 的信息素浓度, μ 表示信息素局部挥发因子, Q_1 为局部更新常量。

通过式(10)降低了 (j, r) 的信息素浓度, 一方面降低了其它蚂蚁对该链路访问的可能性, 减少了无效搜索; 另一方面随着迭代次数的增加, 将会使蚂蚁的搜索范围限制在以源点为中心, 以该类链路为边界的区域内, 提高了搜索效率。由于死点并非叶子节点, 式中参数选取不应使链路信息素急剧减少。

3.3 全局信息素更新规则

全局信息素更新要能引导后代蚂蚁在优秀解周围搜索, 提高算法速度, 同时保证算法的搜索能力。在完成当前循环多播树合并后按下式更新所有链路信息素:

$$\tau_{jr}^{c+1}(0) = (1 - \rho)\tau_{jr}^c(\infty) + \rho\Delta\tau_{jr} \quad (11)$$

式中 $\tau_{jr}^{c+1}(0)$ 表示第 $c+1$ 次循环时链路 (j, r) 的初始信息素值; $\tau_{jr}^c(\infty)$ 表示第 c 次循环结束时链路 (j, r)

r) 的信息素值 ρ 表示信息素全局挥发因子; $\Delta\tau_{jr}$ 表示经过当前循环链路 (j, r) 信息素变化量。

设当前循环中共生成了 N_T 棵多播树, 根据式(5)将这些多播树按费用由小到大进行排列得 $RT_1, RT_2, \dots, RT_{N_T}$, 选择前 ϑ_1 棵多播树, $\forall RT_i, 1 \leq i \leq \vartheta_1$ 按式(12)计算其链路信息素增量。

$$\Delta\tau_{jr} = \Delta\tau_{jr} + \alpha(e) \cdot \frac{Co_Best}{Co(RT_i)} \cdot Q_2 \quad (12)$$

其中 Co_Best 表示到当前循环为止多播树的最低费用; Q_2 为信息素更新常量;

$$\alpha(e) = \begin{cases} 0, & Z^+(e) = Z^-(e) \\ \sqrt{\max_{\bar{e} \in E^*} \{R(\bar{e})\}}, & \text{else} \end{cases} \quad (13)$$

选择排列中后 ϑ_2 棵多播树, $\forall RT_{N_T-i+1}, 1 \leq i \leq \vartheta_2$ 按式(14)计算其链路信息素增量。

$$\Delta\tau_{jr} = \Delta\tau_{jr} - \frac{Co_Best}{Co_CurBest} \cdot Q_3 \quad (14)$$

其中 Co_Best 表示当前循环的最低多播树代价; Q_3 为信息素更新常量。

式(11)、式(12)奖励了较优多播树, 提高了其链路的信息浓度, 使下次循环蚂蚁倾向于在部分较优解周围进行, 提高了算法速度。相反, 式(14)则惩罚了较劣多播树, 一方面减小了蚂蚁访问该类多播树链路的可能性, 另一方面则提高了未被访问链路被搜索的机率, 增强了算法的全局搜索能力。

3.4 基于蚁群算法的多播问题求解步骤

利用本文蚁群算法求解网络 QoS 多播路由步骤如下:

Step 1 删除网络中不满足丢包率约束节点, 以及不满足带宽约束链路。

Step 2 初始化链路信息素浓度, 并设置相关算法参数 $\alpha, \beta, q_0, Q_1, \mu, Q_2, \rho, Q_3, \vartheta_1, \vartheta_2$ 以及蚂蚁子群规模 m 和最大循环次数 C_{max} ; 初始化循环次数 $c = 0$ 。

Step 3 设置当前循环次数 $c = c + 1$ 并设置各链路 $\Delta\tau_{jr} = 0, \rho = 0$; 将 $m \times |D|$ 只蚂蚁置于源点, 为每只蚂蚁生成禁忌表, 并将源点放入禁忌表中。

Step 4 分解多播树。 $t = t + 1$, 依次从各子群中选择一只未完成搜索的蚂蚁根据状态转移规则选择下一节点 r , 如果不存在 r 则根利用局部信息素更新规则调整相应链路信息素。否则将 r 放入蚂蚁禁忌表, 如果 r 为对应子群的目标点, 则标识该蚂蚁完成搜索。

重复本步骤直到所有蚂蚁完成搜索。

Step 5 根据多播树合并策略生成 γ 棵多播树, 并根据全局更新规则调整信息素, 记录到当前为止的最优多播树 RT_{best} 。如果 $c < C_{max}$ 则执行 Step 3, 否则实行 Step 6。

Step 6 输出最优多播树, 算法结束。

4 仿真实验

仿真实验网络拓扑结构生成参照文献[6]中方法, 各项 QoS 参数设置借鉴文献[5]中的均匀随机方法。实验中多播源点和目标点随机选择, 实验主要验证算法的收敛速度和寻优能力。

表1、表2描述了在不同网络规模下, 本文算法与文献[5]中启发式遗传算法, 文献[7]中 PSO 算法的收敛速度对比。从中可以看出由于蚁群算法的正反馈机制, 加之本文的信息素更新策略, 算法体现出较快的收敛速度, 且不随网络规模的增大而急剧变化, 保持了良好的特性, 适合对大型网络的应用。

图2显示了算法在网络规模 $|E| = 20, 70$ 时的两个仿真实例搜索过程。从图中可见, 算法在收敛到最终解之前陷入了局部极值, 但在多播树合并策略控制下, 跳出了局部最优解, 并朝向更优解的方向搜索, 最终以较快速度收敛于全局(近似)最优解。

表1 算法收敛次数对比

Tab.1 Comparison of convergent times

网络规模	遗传算法/次	本文算法/次
20	10	12.3
40	29	27.875
80	41	37.778
100	73	61.585
120	112	81.012

表2 算法收敛时间对比

Tab.2 Comparison of convergent time

网络规模	PSO 算法		本文算法	
	P4	2.4G 256M	P4	2.0G 256M
20	0.11		0.077	
40	0.20		0.185	
60	0.46		0.330	
80	1.35		0.671	
100	1.75		0.953	

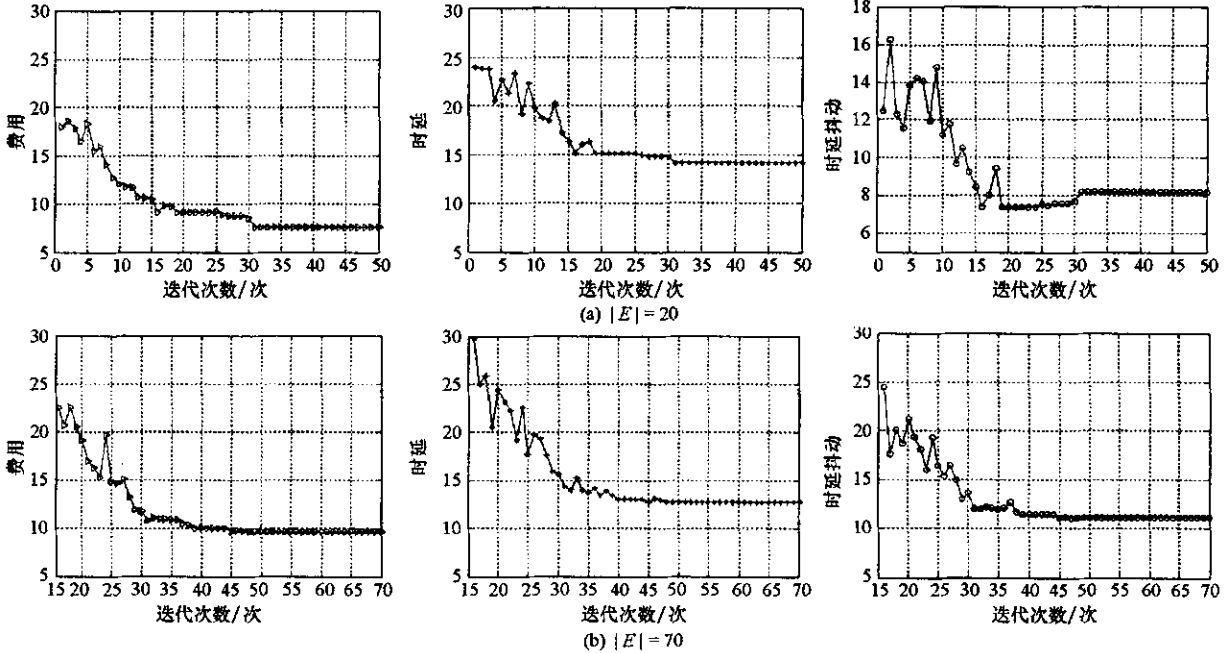


图2 不同网络规模算法搜索过程

Fig.2 The process of algorithm for different network size

5 结论

本文在研究 QoS 多播问题的基础上,提出了一种基于蚁群算法的方法对该问题进行求解。该方法通过将多播问题分解为对单目标的单播问题,并在合并控制策略下重组多播树,使算法能从局部最优解中跳出,提高了全局搜索能力。针对蚁群中蚂蚁搜索可能失败的情况,在蚁群算法中还引入局部信息素惩罚更新机制,缩小了搜索空间,提高了搜索效率,并在全局更新阶段引入了基于多播树奖惩的更新机制,提高了算法的收敛速度。仿真实验表明,该方法能有效地求解 QoS 多播路由问题。

参考文献:

[1] WANG Z, et al. Quality-of-Service Routing for Supporting Multimedia Application[J]. IEEE JSAC, 1996, 14(7):1228-1234.
 [2] Rocha M, Sousa P, Rio M, Corez P. QoS Constrained Internet Routing with Evolutionary Algorithms[C]//2006 IEEE Congress on Evolutionary Computation, 2006 2720-2727.
 [3] 屈建伟,李腊元.一种混合蚂蚁算法及其在 QoS 多播路由中的应用[J]. 武汉理工大学学报,2005,29(6):932-935.
 [4] 顾军华,候向丹,等.基于蚂蚁算法的 QoS 多播路由问题求解[J]. 河北工业大学学报,2002,31(4):19-24.
 [5] 王征应,石冰心.基于启发式遗传算法的 QoS 多播路由问题求解[J]. 计算机学报,2001,24(1):55-61.
 [6] Salama H F, Reeves D S, Viniotis Y. Evaluation of Multicast Routing Algorithms for Real-time Communication on High-speed Networks [J]. IEEE JSAC, 1997, 15(3) 332-245.
 [7] 潘达儒,杜明辉.基于粒子群优化的 QoS 多播路由算法[J]. 计算机工程与应用 2006 42(1):138-140.
 [8] 江泽焱.一种基于多目标优化的 QoS 路由交互式算法[J]. 国防科技大学学报,2002,24(4):37-41.
 [9] Oliveria C A S,Paradalos P M. A Survey Fcombinatorial Optimization Problems in Multicast Routing[J]. Computer & Operations Research, 2005, 1935-1981.

