

基于混合遗传算法的多约束组播路由问题的求解^{*}

谢黎明¹, 余丰人², 丘海明²

(1. 广东轻工职业技术学院, 广东 广州 510300;
2. 中山大学电子与通信工程系, 广东 广州 510275)

摘 要: 研究了延时、延时抖动约束的最小费用组播路由问题, 并且提出了一种混合遗传算法。遗传算法是一种模拟生物进化过程的并行最优算法, 适合在大型、复杂的空间中寻找最优解。遗传算法并不要求解问题具有连续性, 因而适用于约束组播路由问题的求解。此外, 遗传算法的并行分布式处理实现简单, 因而用遗传算法解决约束组播路由问题十分有效。并通过计算分析, 表明本算法能够快速有效的构造出所要求的满足延时、延时抖动约束的最优组播树。

关键词: 混合遗传算法(HGA); 组播; QoS; 模拟退火(SA); 指导初始群体生成; 启发式交叉操作

中图分类号: TP30 1 **文献标识码:** A1 **文章编号:** 0529-6579(2005)02-0045-04

随着网络技术的不断发展和 Internet 的迅速普及, 网络正以前所未有的速度发展, 网络规模将进一步扩大, 网络信息流量迅速增加, 从而网络将变得更加拥挤, 这将严重影响网络的传输速率。为了减少网络上传输信息包的总量, 提高网络的传输能力, 组播技术在网络传输上得到了广泛应用。组播技术要解决的关键问题之一就是组播路由问题。Kompella 等^[1] 在 1992 年首次提出了一种满足给定端到端延时约束的组播路由算法, 并证明此类问题是 NP 完全问题。对于约束组播路由的求解, 多年来提出了许多不同的算法, 其中最引人注目的是近几年出现的一种新型优化算法——遗传算法。使用遗传算法求解科学研究和工程技术中各种组合搜索和优化计算问题的这一基本思想, 在 20 世纪 60 年代初由美国 Michigan 大学的 J. Holland 教授提出^[2], 20 世纪 80 年代中期得到蓬勃发展。但在其发展的同时, 遗传算法也暴露出许多不足和缺陷, 针对其不足, 近年来又有许多改进的方法出现, 其中之一就是将基本遗传算法与传统优化方法相结合的混合遗传算法。本文正是利用混合遗传算法, 对多约束的组播路由问题进行求解。

1 多约束组播路由问题混合遗传算法

采用树结构编码。对于信源 s , 设与之相连的节点数为 N_s , 计算与之相连的 N_s 个边($e_1, e_2, \dots, e_{N_s}$) 的适应度值($f_1, f_2, \dots, f_{N_s}$), 采用概率

$$P_i = \left[f_i + \frac{1}{N_s} \sum f_i \right] \setminus \left(2 \sum f_i \right)$$

不断连接邻节点, 最后删除不是信宿的叶子节点, 从而获得一个初始个体。重复这一步骤, 直到初始个体数目满足要求为止。本文算法中, 采用的群体规模保持不变, 取值为 15。

对于组播树有 $T = (V_T, E_T)$, ($T \subseteq G$)
个体费用为 $f_c = \text{Cost}(T) = \sum_{(a,b) \in E_T} (a, b)$

个体最大延时为
 $f_{D \max} = \max \{f_{Dv1}, f_{Dv2}, \dots, f_{DvN}\}$

个体最大延时抖动为
 $f_{DJ \max} = \max \{f_{DJv1}, f_{DJv2}, \dots, f_{DJvN}\}$

其中,

$$f_{Dvi} = \sum_{(a,b) \in P_T(s,vi)} DJ(a,b),$$
$$f_{DJvi} = \left| \sum_{(a,b) \in P_T(s,vi)} DJ(a,b) \right|$$

构造适应度函数 $f_T = 1 / (af_c + bf_{D \max} + cf_{DJ \max})$

可见, 此适应度函数值大于零, 在本文算法示例中 a, b, c 的取值一律取 1。

选择方法采用最佳个体保留法, 即每代的适应度值最高的个体直接进入下一代, 其余个体是否直接进入下一代则采用比例选择方法, 第 i 个个体 T_i 的被选择概率为:

$$P_{si} = f(T_i) / \sum f(T_i)$$

在进化后期, 由于产生的群体可能适应度相差

^{*} 收稿日期: 2004-09-01
基金项目: 国家自然科学基金资助项目(60472010)
作者简介: 谢黎明(1965年生), 男, 讲师; 通讯联系人: 丘海明; E-mail: xlm@gdqv.edu.cn
(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

极小, 为此结合模拟退火方法, 对适应度进行拉伸, 拉伸方法如下:

$$f'_i = (e^{f_i/T} - 1) \backslash \left(\sum_{i=1}^M e^{f_i/T} - M \right)$$
$$T = T_0 (\alpha^{g-1}) \quad (0 < \alpha < 1)$$

T_0 为初始温度, T_0 和 α 随具体问题而变。适应度的拉伸作用, 要求保证在初期拉伸后适应度相差不大; 而后期为了使更优良个体突出, 要求各个体适应度拉伸后的差距变大。本算法中适应度拉伸采用的 α 值为 0.8, 初始温度 T_0 采用初始群体平均适应度值 ($\frac{1}{15} \sum f_{i \text{ begin}}$)。

交叉操作是从当代群体中随机地选择两个个体 (这两个个体可以包含选择策略中保存到下一代的个体), 依照下述的交叉规则以概率 1 交叉产生一个新子女个体^[3]。该过程一直重复到下一代个体数与初始群体数目相同为止。交叉规则如下: ①选择父母中相同的链路, 将其直接遗传到子女。下一步就是连接这些零散子树成为一棵完整的树。②连接这些零散子树前首先对这些零散子树进行分类, 第一类子树包括源节点, 记为 T_s ; 第二类子树包括目的节点, 记为 T_{vi} ; 第三类子树即不包括源节点, 也不包括目的节点, 记为 T_{mi} 。

组成完整组播树连接规则采用如下启发式的连接方法: 如果所选择的父母个体在延时或延时抖动约束上只有一项不满足, 则在子树间采用使该项约束最小的最短路径连接, 若父母个体 QoS 约束全不满足或各满足一个或全满足, 则子树间随机用“最短延时路径”、“最小延时抖动路径”和“最小费用路径”连接。

交叉完成后, 新的子女个体以概率 P_m 进行变异, 变异概率取 0.05^[3]。变异规则如下: 从新子女个体中, 随机选择一些中间节点 (非组播源节点和目的节点), 删除连接这些节点的路径形成零散子树, 然后依照与交叉操作相类似的连接方法连接各零散子树, 但“最短延时路径”、“最小延时抖动路径”和“最小费用路径”随机选择。

本文采用如下的算法结束条件: 当相同的最优个体在群体中占据的比例大于时, 结束算法操作, 最终结果为算法最后一代中的最优个体。

2 实验及其分析

采用美国州际通信网进行实验^[4], 拓扑图见图 1, 该网共有 28 个节点、45 条边。拓扑图上的数据 (d, d_j, c) 采用文献 [5] 的数据。算法采用 C 语言编程对含 QoS 约束的组播路由进行计算。实

验主要验证算法的收敛性、收敛速度和收敛过程。

文献 [6] 证明, 具有变异概率 $P_m \in (0, 1)$, 交叉概率 $P_c \in [0, 1]$, 同时采用比例选择和最优个体保留的遗传算法可收敛到全局最优解。而本文提出的遗传算法具有以下特点: ①以概率 1 进行交叉; ②变异概率为 0.05; ③最优个体保留和比例选择。故本文提出的混合遗传算法定能收敛到全局最优解。本算法计算了从节点 0 到节点 {9、11、18、21、27} 的组播路由, 要求延时小于 90 ms, 延时抖动在 20 ms 范围内。迭代采用的群体数为 15, 适应度拉伸采用的 α 值为 0.8, 初始温度 T_0 采用初始群体平均适应度值 ($\frac{1}{15} \sum f_{i \text{ begin}}$), 本文取值为 0.001 464。实验的平均收敛代数数为 8 代, 收敛结果如图 1 所示。表 1 是算法计算中的一次实例数据。

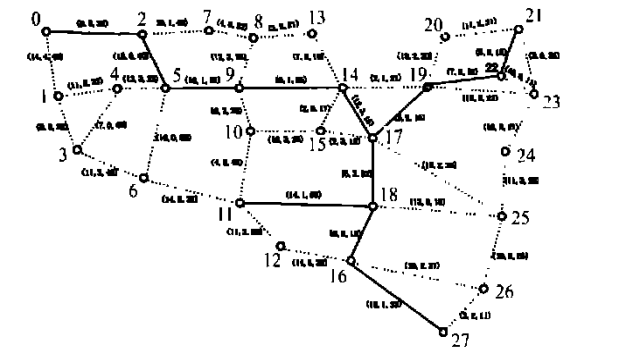


图 1 美国国际通信网中节点 0 到节点 {9、11、18、21、27} 的最佳路由

Fig 1 An optimal router of U. S. interstate communication network

3 结 论

本文对延时、延时抖动约束的最优组播路由问题进行了研究, 提出了一种性能良好的混合遗传算法。实验结果表明此混合遗传算法的性能良好, 收敛速度很快。本文针对基本遗传算法中存在的不足, 采用混合遗传算法对其进行改进, 主要是在基本遗传算法的基础上加入初始群体指导生成、模拟退火算法对适应度拉伸以及采用启发式算法进行交叉连接。初始群体的指导生成目的是为了使初始群体具有一定的优良性, 从而使算法从第 1 代开始就能在较好的搜索空间中进行搜索, 从而加快算法的收敛速度。

表 1 本算法计算中的一次实例数据

Tab 1 An example data

编号	1 代	2 代	3 代	4 代
1	(161, 29, 441)✓	13(109, 21, 408)✓	13(74, 15, 344)	11(74, 15, 333) ☆
2	(145, 29, 470)	1(161, 29, 441)	1(109, 21, 408)	4(103, 18, 396)
3	(139, 26, 515)	5(166, 24, 588)	11(98, 25, 454)	7× 12(71, 14, 371)
4	(182, 27, 566)	9(168, 32, 520)	15(103, 18, 396) ✓	11× 14(76, 14, 370)
5	(166, 24, 588)✓	10(201, 37, 595)	1× 10(87, 14, 383)	5× 9(74, 15, 339)
6	(130, 20, 466)	8× 15(121, 20, 533)	2× 13(76, 14, 370)	9× 13(82, 12, 396)
7	(134, 28, 407)	1× 13(79, 16, 384)	8× 12(74, 15, 338)	5× 10(76, 14, 483)
8	(151, 28, 532)	3× 5(74, 15, 457)	4× 8(85, 15, 482)	4× 7(74, 15, 338)
9	(168, 32, 520)✓	7× 14(95, 15, 360)	13× 15(74, 15, 344)	3× 6(71, 14, 371)✓
10	(201, 37, 595)✓	5× 7(76, 14, 370)	3× 11(98, 25, 414)	6× 12(71, 14, 357)
11	(192, 36, 588)	3× 9(98, 25, 454) ✓	7× 8(74, 15, 333) ☆	3× 8(86, 18, 455)
12	(166, 30, 523)	10× 11(74, 15, 359)	7× 9(68, 11, 371)	3× 13(76, 14, 365)✓
13	(109, 21, 408) ☆	3× 14(74, 15, 344) ☆	5× 7(76, 14, 374)	8× 9(82, 19, 331)
14	(134, 21, 431)	6× 7(87, 14, 411)	6× 14(90, 12, 362)	1× 13(76, 14, 365)
15	(183, 32, 610)	9× 12(103, 18, 396)✓	10× 13(74, 15, 344)	3× 9(69, 15, 345)
编号	5 代	6 代	7 代	8 代
1	1(74, 15, 333) ☆	1(74, 15, 333) ☆	1(74, 15, 333) ☆	1(74, 15, 333) ☆
2	9(71, 14, 371)	6(74, 15, 333) ☆	2(74, 15, 333) ☆	2(74, 15, 333) ☆
3	12(76, 14, 365)	12(74, 15, 333) ☆	3(74, 15, 333) ☆	3(74, 15, 333) ☆
4	8× 13(69, 15, 339)	2× 13(69, 15, 339)	5(74, 15, 333) ☆	4(74, 15, 333) ☆
5	5× 6(82, 14, 349)	12× 15(74, 15, 333) ☆	6(74, 15, 333) ☆	5(74, 15, 333) ☆
6	12× 13(74, 15, 533) ☆	4× 12(74, 15, 333) ☆	9(74, 15, 333) ☆	6(74, 15, 333) ☆
7	1× 6(82, 19, 331)	11× 15(76, 14, 364)	10(74, 15, 333) ☆	7(74, 15, 333) ☆
8	2× 6(82, 12, 384)	11× 13(69, 15, 339)	11(74, 15, 333) ☆	8(74, 15, 333) ☆
9	10× 11(76, 14, 483)	3× 6(74, 15, 333) ☆	3× 12(74, 15, 338)	11(74, 15, 333) ☆
10	4× 15(69, 15, 345)	6× 7(74, 15, 333) ☆	2× 15(69, 15, 339)	9× 12(74, 15, 338)
11	7× 14(76, 14, 364)	6× 14(74, 15, 333) ☆	9× 15(74, 15, 333) ☆	12× 13(74, 15, 338)
12	1× 14(74, 15, 333) ☆	1× 15(74, 15, 338)	12× 13(69, 15, 339)	6× 14(74, 15, 338)
13	8× 15(69, 15, 339)	2× 11(71, 14, 365)	5× 12(74, 15, 338)	2× 4(74, 15, 333) ☆
14	1× 4(74, 15, 338)	9× 10(74, 15, 338)	2× 4(69, 15, 339)	8× 9(74, 15, 333) ☆
15	2× 4(76, 14, 370)	8× 9(69, 15, 339)	2× 14(74, 15, 338)	4× 12(69, 15, 339)

1) ☆表示当代中最优个体, ✓表示经比例选择选中而被保留的个体。每个个体用 (d, d_j, c) 表示性能, 左侧数据表示此个体来源, 如 12 表示此个体由上代个体 12 直接保留而来, 4× 15 表示此个体由上代 4, 15 个体交叉生成。

参考文献:

[1] KOMPELLA V P. Multicasting for multimedia applications [C] . IEEE INFOCOM' 92, 1992: 2078— 2085.

[2] HOLLAND J H. Adaptation in nature and artificial systems [M] . Michigan: The University of Michigan Press, 1975.

[3] 王征应. 基于启发式遗传算法的 QoS 组播路由问题的求解 [J] . 计算机学报, 2001(1): 55— 61.

[4] WINTER P. Steiner Problem in Networks: A Survey[C] . Networks, 1987: 129— 167.

[5] 吴新余. Internet 中的多播路由选择算法 [J] . 南京邮电学院学报(自然科学版), 1999(2): 1— 3

[6] 陈国良. 遗传算法及其应用[M] . 北京: 人民邮电出版社, 1996.

[7] HWANG F K. Steiner tree problems[C] . Networks, 1992: 55— 89.

[8] ZHU Q. A source based algorithm for delay-constrained minimal cost multicasting[C] . Proc IEEE INFOCOM' 95, 1995: 377— 384.

Solution of QoS Multicast Routing Problems Based on HGA

XIE Li ming¹, YU Feng ren², QIU Hai ming²

(1. Guangdong Industry Technical College, Guangzhou 510300, China;

2. Department of Electronics and Communication Engineering, Sun Yat sen University, Guangzhou 510275, China)

Abstract: A hybrid genetic algorithm (HGA) was presented to solve the delay and delay jitter constrained least cost multicast routing problem. Genetic algorithm (GA) is a kind of parallel optimization algorithm that simulates the evolution process of a creature, and is suitable for finding optimal solution in a large and complicated search space. GA does not need continuous solution to the problem, and can be easily implemented in parallel distributed process, so that it is effective in solving QoS multicast routing.

Key words: hybrid genetic algorithm (HGA); multicast; QoS; simulated annealing algorithm (SA); instructional preliminary colony building; heuristic crossover operation

(上接第 41 页)

A Robust ARMA System Identification Based on the Vector Method

HU Xue jun, ZHAO Hui min, CHEN Guang yi

(Foshan University, Guangdong Foshan 528000, China)

Abstract: To present a new approach to autoregressive and moving average (ARMA) modeling based on the support vector method, a statistical analysis of the characteristics of the proposed method is carried out. The results show, compared SVM-ARMA with LS, precisions of validation prediction error of the SVM-ARMA improved 1.2 dB than LS in some conditions. Besides, the effect of outliers can be cancelled. With using nonlinear kernels, the system will lead to robust, nonlinear system identification procedures.

Key words: ARMA modeling; support vector method; system identification; times series