

文章编号: 1000-6788(2003)05-0103-05

求解度限制最小生成树问题的启发式遗传搜索算法

王励成, 孙麟平

(南京大学数学系, 江苏 南京 210093)

摘要: 提出了求解度限制最小生成树问题的启发式遗传搜索算法, 通过大量的数值试验验证其可行性; 最后对此算法做了适当推广, 并给出了用它求解 TSP 及 MTSP 等问题的具体步骤和实例。

关键词: 遗传算法; 启发式搜索; 度限制最小生成树

中图分类号: O 224

文献标识码: A

A Heuristic Genetic Search Algorithm for Solving the Degree-Constrained Minimum Spanning Tree Problem

WANG Li-cheng, SUN Lin-ping

(Mathematics Department, Nanjing University, Nanjing 210093, China)

Abstract We propose a heuristic genetic search algorithm for solving the degree-constrained minimum spanning tree problem. The feasibility of the algorithm is illustrated by amount of numeric experiments. Finally, the extent of the algorithm is expanded properly. We also present the details and some examples of solving the travelling salesman problems and the multiple travelling salesman problem with this algorithm.

Keywords genetic algorithm; heuristic search; degree-constrained minimum spanning tree

1 引言

度限制最小生成树 (Degree-Constrained Minimum Spanning Tree) 问题可以看作是 MST 问题的进一步抽象, TSP 和 MTSP 等问题均可看作是 DCMST 问题的特例, 而前者有更多的应用背景^[1,2], 故研究其求解方法有现实的意义。文献[1]是先生成 MST (采用 Prime 或 Kruskal 算法), 再根据度约束条件对其进行修剪换枝 (采用分支定界算法), 故时间复杂度较高 (因为分支定界算法本身的时间复杂度是指数级的)。文献[2]采用蚂蚁算法进行搜索, 这不失为求解高复杂度问题的明智选择; 但其模型中对生成树的约束描述采用代数方法 (如一组等式或不等式), 这种描述较为复杂, 要计算节点集合的所有子集^[2], 时间复杂度为 $O(2^n)$, 不易程序实现。广度优先搜索算法在求一个图的生成树问题上是十分简便的, 我们尝试用这一过程性算法代替复杂的代数方法。

如果在搜索过程中将边权考虑在内, 就可较快找到最小生成树, 这正是传统启发式搜索的基本思想; 但是, 这种启发式算法 (又称瞎子爬山法) 容易陷入局部极小^[3], 对于 DCMST, 它还不能保证找到有效解, 故考虑别的启发信息。由于含 n 个节点的完全图共有 n^{n-2} 棵不同的生成树, 面对如此巨大的搜索空间, 我们自然而然想到了遗传算法^[4], 即利用其隐含的并行搜索能力, 让一个染色体代表寻找一棵生成树的启发信息, 再运行启发式搜索得到一棵生成树; 当种群数量为 P 时, 第 k 代就可得到 P 棵生成树, 然后根据它们的树权进行评价, 进而可通过选择、杂交和变异等遗传操作得到第 $k+1$ 代的 P 棵生成树, 并在进化过程中始终保留当前最优个体, 就可依概率 1 找到 DCMST 的全局最优解^[5,6]。

2 算法设计

先给出启发式搜索算法的描述 (以广度优先搜索算法为基础), 并由此来推导一个染色体上各等位基

收稿日期: 2000-04-04

因的含义, 然后给出遗传算法设计及数值试验结果 .

2 1 带有遗传信息的启发式搜索算法(Heuristic Search with Genetic Information)

设原DCM ST 问题的参数为

$$W = (w_{ij})_{n \times n}$$

边权矩阵

$$b = (b_i)_{1 \times n}$$

度限制向量

并设一染色体为

$$a = (a_1, a_2, \dots, a_n)$$

它表示求解一棵生成树的遗传指导信息,HSGI算法的基本思想是: 第 i 步, 挑选与当前生成树相邻的但尚不在生成树中的第 a_i (取模意义下) 条边进入生成树 . 算法简述如下:

算法名: HSGI(a)

[算法开始]

- 1) 以第 a_1 个节点为生成树 T 的根 ;
- 2) 设第 i 步后, 已在 T 中的节点集合为 $V_T = \{v_1, v_2, \dots, v_i\}$;
- 3) 计算 T 之外与 V_T 中的节点相邻的边的集合 B , 记 $|B| = m$, 并对这 m 条边规定任意一种次序关系;
- 4) 若 $m > 0$, 则 将 B 中第 a_{i+1} 条边加入到生成树中 ; 否则, 转 6);
- 5) $i = i + 1$, 重复 2)- 4);
- 6) 若 n 个节点都在 T 中, 则 T 就是由染色体 a 启发搜索出的生成树, 算法正常退出; 否则, 原图不连通, 不存在生成树, 算法异常退出 .

[算法结束]

2 2 遗传算法设计

基因编码为

$$a = (a_1, a_2, \dots, a_n) \quad N^n$$

随机产生;

一致杂交算子如表 1 所示:

表 1 一致杂交算子

父代染色体 1	$(a_1, a_2, \dots, a_i, \dots, a_n)$
父代染色体 2	$(b_1, b_2, \dots, b_i, \dots, b_n)$
随机生成的杂交控制串	$(0, 1, \dots, 1, \dots, 0)$
子代染色体 1	$(a_1, b_2, \dots, b_i, \dots, a_n)$
子代染色体 2	$(b_1, a_2, \dots, a_i, \dots, a_n)$

基因变异算子: 某个基因位变异时加上一个随机整数; 染色体翻转变异算子如表 2 所示(翻转起止位 $1 \leq i < j \leq n$ 随机生成)

表 2 染色体翻转变异算子

变异前	$(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{j-1}, a_j, a_{j+1}, \dots, a_n)$
翻转起止位	
变异后	$(a_1, a_2, \dots, a_{i-1}, a_j, a_{j-1}, \dots, a_{i+1}, a_i, a_{j+1}, \dots, a_n)$

在模 n 意义下

对于完全图, $B = V_T \times (V_T)^c$, 其中 $(V_T)^c$ 为 V_T 的补集 但实际计算时, 每次只对 V_T 和 B 做简单校正, 计算量已大大减小 .

模 m 意义下 .



适应度函数: 设 $T = \text{HSGI}(a)$, 则

$$f(a) = [w(T) + P(T)]^{-1} * C_{scale}, \quad w(T) = \sum_{\substack{\text{边}(i,j) \\ T}} w_{ij}$$
$$P(T) = \max_{i,j} \{w_{ij}\} * \text{不满足度约束的节点个数};$$

其中调比因子 $C_{scale} = 10000$;

选择策略由赌盘选择和保优措施组成.

给定种群数量 P , 杂交概率 p_c , 基因变异概率 p_m , 染色体翻转概率 p_r , 最大进化代数 $\text{max}G$ 后, 遗传算法可描述为

算法名: $\text{GA_HSGI}(W, b, P, p_c, p_m, p_r, \text{max}G)$

[算法开始]

- 1) 随机生成 P 个初始染色体, $k = 1$;
- 2) 对每个染色体 a 调用算法 $\text{HSGI}(a)$ 得到生成树 $T(a)$, 并计算其适应值 $f(a)$;
- 3) 记录到目前为止的最优个体 $\text{opt_}a$ 及最优解 $\text{opt_}T$;
- 4) 执行赌盘选择;
- 5) 执行交叉算子, 基因变异算子和染色体翻转算子;
- 6) $k = k + 1$;
- 7) 重复 2) 至 6), 直到 $k = \text{max}G$;
- 8) 返回 $\text{opt_}a$ 及 $\text{opt_}T$.

[算法结束]

2.3 数值试验及结果分析

对文献[1]和[2]中给定的实例, 分别设置不同的控制参数, 运行 GA_HSGI 算法 10 到 20 次, 部分较好的结果如表 3、表 4 所示; 分别选取两例某一次运行的适应值攀升轨迹如图 1 和图 2. 可见, 1) 在 50 代之前, 算法性能十分良好, 种群平均适应值和最大适应值几乎呈指数上升; 2) 而为了以较高的概率找到全局最优解, 设置较大的变异率和足够大的进化代数是必要的, 这同文献[7]中的观点一致.

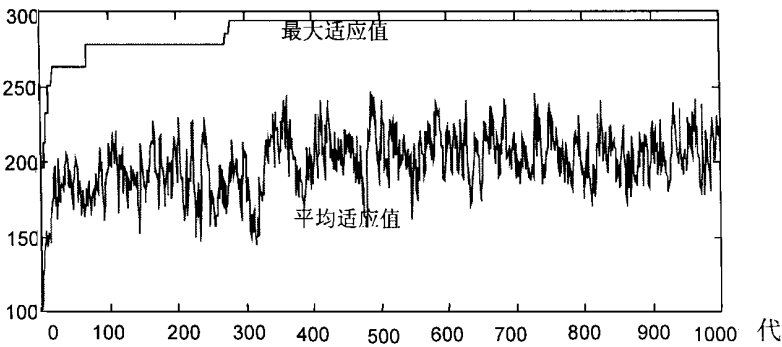


图 1 实例 1 的某一次运行

表 3 对文献[1]中实例的数值试验

P	P_c	p_m	p_r	$\text{max}G$	最优解/有效解/运行次数	$E[w(T)]$
30	0.5	0.05	0	100	13/19/20	35.1
30	0.5	0.05	0	500	14/20/20	34.8
30	0.5	0.05	0.005	1000	19/20/20	34.1
30	0.3	0.08	0.008	2000	13/13/13	34.0

文献[1]中给出的最优解为 34.

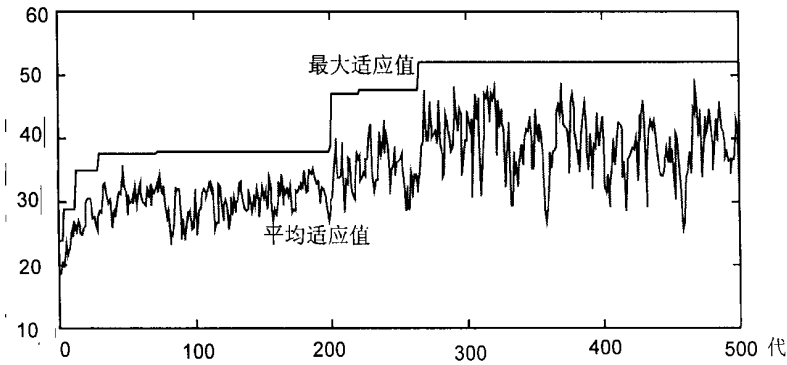


图 2 实例 2 的某一次运行

表 4 对文献[2]中实例的数值试验

P	P_c	p_m	p_r	$\max G$	最优解/有效解/运行次数	$E[w(T)]$
10	0.2	0.05	0.005	100	0/7/10	212.28
10	0.2	0.05	0.005	500	2/7/10	189.86
10	0.2	0.05	0.05	500	0/8/10	202.875
10	0.2	0.05	0.005	1000	4/7/10	178.9

特别地,对于文献[2]中的实例,我们的算法不仅多次达到原文中的最优值 192,而且也多次找到新的更好的最优值 169,其对应的解如图 3 所示(括号内为边权)。

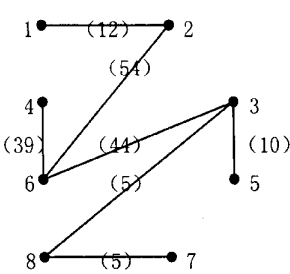


图 3

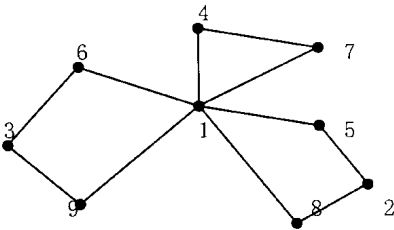


图 4

3 算法的推广

3.1 一般图上的最小生成树或最小生成森林的求解

注意到 2 中所述算法并不要求 G 一定是完全图,而是基于一般连通图 G 的.当 G 不连通时:1) 在 G 的 k 个的连通分支之间添加 $k-1$ 条权为 0 的边得到连通的图 G_1 ,并修改相应节点的度约束(如图 5);2) 再用算法 GA_HSGI 求得 G_1 的度限制最小生成树 T ;3) 从 T 中删除第 1) 步所添加的边,即得到 G 的度限制最小生成森林 F .

3.2 TSP 问题的求解

用 GA_HSGI 算法求解 TSP 问题时,选择距离最近的两个城市 A 和 B ,并置 A, B 两节点的度限制为 1,其余节点的度限制为 2,再删除边 (A, B) ,即将 A, B 两城的距离修改为无穷大.这时就可此算法求得以 A, B 为叶子节点的线性树,最后重新加入边 (A, B) 就得原 TSP 问题的最优解.对于典型的 Hopfield 的 10 城 TSP 问题^[8],分别设置 $P=20, P_c=0.2, p_m=0.05, p_r=0.005$ 或 $0, \max G=500$ 或 1000 ,用此算法找到了最优解 $L(B-C-A-D-E-F-G-H-I-J-B)=26908$,它优于文献[8]中给出的两个最优解 271 和 283.

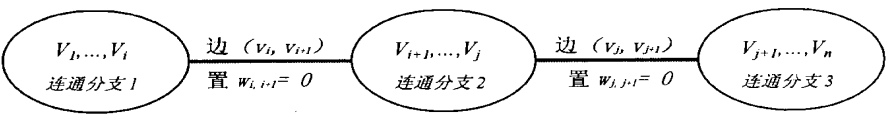


图 5

3 3 M TSP 问题的直接求解法

对 M TSP 问题, 可通过增加虚拟城市节点将其转化为 TSP 问题(这正是传统的做法, 会引起问题维数的增加), 再用 3 2 中的方法求解; 也可利用本文提出的算法直接进行求解(不会引起问题维数的增加). 例如, 有一 M TSP 问题描述为: m 个旅行商, 出发地均为 A 城, 总共要访问 n 个城市, 权矩阵为 W . 首先, 将此问题映射到一个由 W 定义的无向完全图上并在其上寻找 m 条边权最小、互不相邻且均与节点 A 不相关的边 $(u_1, v_1), \dots, (u_m, v_m)$, 删除它们(即将权值修改为无穷大), 定义节点 A 的度限制为 $2m$, 节点 $u_1, v_1, \dots, u_m, v_m$ 的度限制均定义为 1, 其余节点的度限制均定义为 2; 这时, 就可用 GA_HSGI 算法求得满足度限制的最小生成树 T ; 最后重新加入边 $(u_1, v_1), \dots, (u_m, v_m)$ 就得原 M TSP 问题的解.

为测试此算法求解 M TSP 问题的能力, 我们特意设计如图 4 所示用例, 这是一个 9 城市 3 旅行商问题, 并使它具有两个极难求解的特点: 1) 使全局最优解 $L(1-4-7-1-5-2-8-1-6-3-9-1)=11$ 是唯一的, 这通过设置 L 中的边权均为 1, 而其余边权均大于 1 来体现; 2) 为寻找全局最优设置很强的干扰, 这通过设置在 L 中的边权较接近于 1 来体现(我们设置为 $[1, 2, 2, 2]$ 区间上的随机值). 通过大量的数值试验($3^25=243$ 组不同参数), 我们发现了两组较好的参数: $P=10, p_c=0.2, p_m=0.5, p_r=0$ 或 $0.005, \max G=1000$, 对每组参数运行 GA_HSGI 算法 5 次, 收敛到 L 的次数分别为 4 和 5, 这不仅说明此算法是有效的, 而且也显示出了染色体翻转变异算子对提高遗传算法全局寻优能力的作用.

为更有效地寻找较好的参数组合, 我们对参数空间进行了 5-因子-3-水平级的正交试验^[10](只需对相互正交的 18 组参数进行), 也得到了同样的结果. 顺便指出, 对于 DCMST 问题, 度限制越紧, 求解就越困难^[2], 而每棵树至少得有两个 1 度节点作为树叶, 故 TSP 及 M TSP 是 DCMST 族中最难的两类.

4 结束语

通过 2 3 中的数值试验和 3 中的推广, 可以看到, GA_HSGI 算法不仅有较好的数值效果, 而且有较宽的适应范围; 综上, 可得以下结论:

- 1) 对于复杂的优化问题, 若有效解的结构比较特殊(如二叉树或生成树等), 其结构约束不易用代数方法表达时, 可采用高效的过程性判别算法, 并将其判别结果与外部罚函数法^[9]结合使用;
- 2) 若寻求最优解的搜索空间很大, 则可考虑采用进化算法如遗传算法、蚂蚁算法等, 利用其并行搜索能力寻找全局最优解;
- 3) 对于特定的问题, 还可以考虑将传统求解算法与进化算法结合使用, 这也可以有两条途径: 用进化算法为传统算法提供指导信息以寻找全局最优(如本文); 用传统算法加速进化过程, 即将传统算法设计为一类变异算子而使其作用在单个染色体上(如各类混合型遗传算法);
- 4) 遗传算法的参数设定目前只能通过试验来确定, 而正交试验^[10]和亚级遗传搜索^[6]是两种可供选择的策略.

参考文献:

[1] 顾立尧. 带有度约束的最小耗费生成树的分支限界算法[J]. 计算机应用与软件, 1989; 6(6): 49- 54
[2] 马良, 蒋巍. 度限制最小树的蚂蚁算法[J]. 系统工程学报, 1999, 14(3): 211- 214

(下转第 112 页)

性关系,其增长的速率远低于 Apriori. 而 Apriori 运行时间随数据库事务数增加而成指数增长.

5 结束语

TAB-FP 挖掘“长”频繁模式时间效率与空间可伸缩性均优于 Apriori. 对于多层布尔型频繁模式、多维多层多数据类型频繁模式的挖掘,可以用扩展的 TAB-FP 来完成. 只要引入多层的层次信息结构或多维的同维互斥信息结构,并在剪裁子树时引入“祖先后代项目互斥或同维项目互斥”的约束,就可以挖掘这两类模式. 由于多层多维问题中存在大量“长”频繁模式,扩展的 TAB-FP 相对于扩展的 Apriori 算法效率更高、可伸缩性更强.

参考文献:

- [1] Agrawal R, Srikant R. Fast algorithms for mining association rules [A]. Proceedings of the 20th International Conference on Very Large Data Bases [C]. USA: Morgan Kaufmann, 1994. 487- 499.
- [2] Christian Borgelt. Apriori: a program to find association rules [EB/OL]. [Http://fuzzy.cs.uni-magdeburg.de/~borgelt](http://fuzzy.cs.uni-magdeburg.de/~borgelt)
- [3] Han J, Fu Y. Discovery of multiple-level association rules from large databases [A]. Proceedings of the 21st International Conference on Very Large Data Bases [C]. USA: Morgan Kaufmann, 1995. 420- 431.
- [4] Srikant R, Agrawal R. Mining generalized association rules [A]. Proceedings of the 21st International Conference on Very Large Data Bases [C]. USA: Morgan Kaufmann, 1995. 407- 419.
- [5] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation [A]. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data [C]. USA: ACM Press, 2000. 1- 12.
- [6] Agrawal R, Aggarwal C, Prasad V V V. A tree projection algorithm for generation of frequent item sets [J]. Journal of Parallel and Distributed Computing, 2000, 61(3): 350- 371.
- [7] Srikant R, Agrawal R. Mining quantitative association rules in large relational tables [A]. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data [C]. USA: ACM Press, 1996. 1- 12.
- [8] Bayardo R J. Efficiently mining long patterns from databases [A]. Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data [C]. USA: ACM Press, 1998. 85- 93.
- [9] Pasquier N, Bastide Y, Taouil R, Lakhal L. Discovering frequent closed item sets for association rules [A]. Proceedings of the 7th International Conference on Database Theory [C]. USA: Springer, 1999. 398- 416.

(上接第 107 页)

- [3] 陈世福, 陈兆乾, 等. 人工智能与知识工程[M]. 南京: 南京大学出版社, 1997, 124- 125, 165- 191.
- [4] Holland J H. Genetic algorithms[J]. Scientific American 1992, (7): 44- 50.
- [5] Winter G, et al. An Introduction on global optimization by genetic algorithms[A]. Algorithms for Large Scale Linear Algebraic Systems[C], Kluwer Academic Publishers, 1998. 343- 367.
- [6] 刘勇, 康立山, 陈毓屏. 非数值并行算法: 遗传算法[M]. 北京: 科学出版社, 1998.
- [7] 侯格贤, 吴成柯. 遗传算法的性能分析[J]. 控制与决策, 1999; 14(3): 257- 260.
- [8] 沈清, 胡德文, 时春. 神经网络应用技术[M]. 长沙: 国防科技大学出版社, 1993, 45- 50.
- [9] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 1997.
- [10] 石磊, 王学仁, 孙文爽. 试验设计基础[M]. 重庆: 重庆大学出版社, 1997.