

文章编号:1007-6735(2012)05-0461-05

图的 Steiner 最小树的竞争决策算法

熊小华¹, 刘艳芳², 宁爱兵²

(1. 上海第二工业大学 计算机与信息学院, 上海 201209; 2. 上海理工大学 管理学院, 上海 200093)

摘要: 图的 Steiner 最小树问题是一个著名的 NP 难题, 在通讯网络、VLSI 等工程实践中有着重要的应用. 在分析图的 Steiner 最小树问题数学性质的基础上, 提出了图的 Steiner 最小树的竞争决策算法. 为了验证算法的有效性, 求解了 OR-Library 中的基准问题, 测试结果表明了算法具有较好的求解效果.

关键词: 竞争决策算法; Steiner 最小树; 图; 降阶

中图分类号: O 223 **文献标志码:** A

Competitive Decision Algorithm for the Steiner Minimal Tree Problem in Graphs

XIONG Xiao-hua¹, LIU Yan-fang², NING Ai-bing²

(1 College of Computer and Information, Shanghai Second Polytechnic University, Shanghai 201209, China;
2 Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: The Steiner minimal tree problem in graphs(GSTP) is a well-known NP-hard problem. Its applications can be found in many areas, such as telecommunication network design, VLSI design, etc. A competitive decision algorithm was developed to solve the GSTP. The mathematical properties of GSTP were analysed, which can be used to scale down the size of original problem and accelerate the algorithm. To assess the efficiency of the proposed competitive decision algorithm, it was applied to a set of benchmark problems in the OR-Library. In terms of computation times, our algorithm clearly outperforms other heuristics for the Steiner problem in graphs, while obtaining better or comparable solutions.

Key words: competitive decision algorithm; Steiner minimal tree problem; graph; reduction

图的 Steiner 最小树 (Steiner minimal tree problem, GSTP) 问题是一个经典的组合优化问题, 该问题由 Hakimi 和 Levin 于 1971 年首次提出,

Karp^[1]在文献中证明了图的 Steiner 树问题在一般情况下是 NP-完全问题. GSTP 问题有着广泛的应用背景, 例如通讯网络的设计、VLSI 电路的设计、生

收稿日期: 2012-05-17
基金项目: 国家自然科学基金资助项目(70871081); 上海市重点学科建设资助项目(S30504)
作者简介: 熊小华(1978-), 女, 副教授. 研究方向: 算法设计、系统工程. E-mail: xing_xiao_hua@163.com

物计算等^[2-4]. 然而由于 GSTP 问题的 NP 难解性, 传统的精确型算法往往存在计算量大、迭代时间长的缺点, 甚至由于超出计算机的处理能力而无法求解, 在工程实践中求解规模较大的问题时, 大多会寻找一个高效可行的启发式算法.

本文拟在分析 GSTP 问题数学性质的基础上, 使用数学性质对问题进行降阶处理, 然后采用竞争决策算法对降阶后的问题进行求解.

1 竞争决策算法简介

竞争决策算法(competitive decision algorithm, CDA)是近几年来提出的一种求解组合优化难题的新型算法^[5-10], 算法的原理详见文献[5]. 自然界中的竞争和决策都是在一定规则下进行, 在竞争者的实力、竞争者和环境间的关系、多个竞争者实力的差距和初始竞争状态等多种因素的共同作用下, 经过多次竞争和决策后, 使不同的竞争者分别占有一定的资源而达到一种新的竞争状态, 只要新的竞争状态优于初始竞争状态, 就能达到优化的目的. CDA 算法吸收了达尔文“优胜劣汰”的进化思想以及演化博弈论中有限理性竞争者的思想, 通过构造一个或多个具有有限理性的竞争者参与到对一个或多个资源的竞争过程中, 通过优胜劣汰的原则使一部分竞争者获得资源而增加实力, 一部分竞争者失去资源, 削弱实力甚至消亡. 当算法通过竞争不能获得更优的结果时, 通过资源交换使算法进入下一轮的竞争. 在理论方面, 现在已给出了竞争决策算法的基本概念和通用流程; 在应用方面, 已利用其通用流程实现了一系列单目标 NP 难题的算法^[6-9]和多目标 NP 难题的算法^[10-11].

2 图的 Steiner 最小树的竞争决策算法

2.1 问题介绍

考虑一个连通的无向简单图(无环无多重边的图就称为简单图) $G = (V, E)$, 其中 $V = \{v_1, v_2, \dots, v_n\}$ 是所有结点的集合, $E = \{e_1, e_2, \dots, e_m\}$ 为所有边的集合. 若边 e_k 的两个端点为 i 和 j , 则边 e_k 可记为 (i, j) , 边 (i, j) 的权值记为 $w(i, j)$ 且为非负实数, 若两结点无边相连则设置 $w(i, j) = \infty$. 给定 V 的一个子集 $P \subseteq V$, 要求在图 G 上寻找一棵子树 $T = (Y, U)$, 使得

$$P \subseteq Y \subseteq V, U \subseteq E$$

且

$$w(T) = \sum_{e \in U} w(e)$$

最小. 则称 T 为图 G 关于 P 的 Steiner 树. 其中, 属于 P 的点称为正则点, 不属于 P 的点称为非正则点, 也称 Steiner 点. 需要强调的是任一棵 Steiner 树必须包含所有的正则点.

2.2 问题的数学性质分析

不同于其它启发式算法, 竞争决策算法具有便于结合问题本身性质的特性. 对问题性质的分析, 可以在一定程度上实现问题的降阶, 这将加快问题的求解速度. 在讨论 GSTP 问题的竞争决策算法之前, 先讨论问题本身的性质.

定理 1 度为 1 的 Steiner 点, 必定不在 Steiner 最小树中.

证明 若 Steiner 点 v_s 的度为 1 且其在 Steiner 最小树上. 此时若把点 v_s 及其关联边从图中删除, 仍然是一棵 Steiner 树且总权值更小, 这与 Steiner 最小树权值最小相矛盾. 因此得证.

由定理 1 可知, 可利用定理 1 对问题进行降阶处理. 在算法最开始, 将所有度为 1 的 Steiner 点从图中移除, 此时可能会产生新的度为 1 的 Steiner 点, 又可利用定理 1 降阶, 直到图中不存在度为 1 的 Steiner 点为止.

定理 2 度为 1 的正则点所关联边, 必定在 Steiner 最小树中.

证明 若度为 1 的正则点 v_t 的关联边不在 Steiner 最小树 T^* 上. 此时必无其它边使 v_t 与 T^* 相连, 此时 T^* 必定不连通的, 这与 Steiner 最小树的连通性相矛盾. 因此得证.

由定理 2 可知, 可利用定理 2 对问题进行再次降阶处理. 在算法最开始, 将所有度为 1 的正则点所关联的边固定下来.

定理 3 度为 2 的 Steiner 点 v_s 所关联结点为 v_x 和 v_y .

a. 若有边 (v_x, v_y) , 且 $w(v_s, v_x) + w(v_s, v_y) > w(v_x, v_y)$, 则正则点 v_s 必定不在 Steiner 最小树中;

b. 若有边 (v_x, v_y) , 且 $w(v_s, v_x) + w(v_s, v_y) < w(v_x, v_y)$, 则边 (v_x, v_y) 必定不在 Steiner 最小树中.

证明 a. 用反证法证. 若 Steiner 最小树包含 Steiner 点 v_s . 由定理 1 可知 v_s 的两条边都在树中, 此时可以将结点 v_s 及其关联边从图中删除, 用更短的一条边 (v_x, v_y) 来替换这两条边且替换之

后仍然是一棵 Steiner 树且总权值更小,这与 Steiner 最小树包含点 v_s 相矛盾,因此得证.

b. 用反证法证,若 Steiner 最小树包含边 (v_x, v_y) . 此时 (v_s, v_x) 和 (v_s, v_y) 必不全在 Steiner 最小树中,否则将形成回路. 此时将 (v_x, v_y) 从图中删除,且让边 (v_s, v_x) 和 (v_s, v_y) 全部在 Steiner 最小树中,此时仍然是一棵 Steiner 树且总权值更小,这与 Steiner 最小树包含 (v_x, v_y) 相矛盾,因此得证.

定理 4 记 v_x 到另一结点 v_y 的最短路权值为 $spm(v_x, v_y)$, 若 $w(v_x, v_y) > spm(v_x, v_y)$, 则边 (v_x, v_y) 必定不在 Steiner 最小树中.

证明 用反证法证,若 Steiner 最小树包含边 (v_x, v_y) . 此时用连接结点 v_x 和 v_y 的最短路上的边替换之,仍然是一棵 Steiner 树且总权值更小,这与 Steiner 最小树包含 (v_x, v_y) 相矛盾,因此得证.

2.3 算法原理

在 GSTP 问题的竞争决策算法中,将图上的所有正则点作为竞争者竞争的资源. 仅有一个非虚拟竞争者 A , 它在算法开始仅占有包含一个正则点的图,其它所有正则点由虚拟竞争者 N 所占有. 竞争者 A 在算法过程中按照竞争力函数和决策原则占领其余的正则点,算法一直运行到竞争者 A 占有所有的正则点.

2.4 基本符号及含义

为了方便描述,本文采用如下符号来表示:

V 为结点集; n 为结点集的元素个数; P 为正则点的集合; m 为正则点元素的个数; $w[i, j]$ 为边 (i, j) 的权值; $arrive[i, j]$ 表示结点 i 与 j 是否直接有边相连; $arrive[i, j] = \text{true}$ 表示边 (i, j) 选中在 Steiner 最小树上; $arrive[i, j] = \text{false}$ 表示边 (i, j) 未被选中; $arrive_t[i, j]: arrive[]$ 的传递对称闭包^[12], 表明结点 i 与 j 是否能够到达(包含通过其它结点中转到达), 若 $arrive[i, k] = \text{true}$ 且 $arrive[k, j] = \text{true}$, 则 $arrive[i, j] = \text{true}$; $spm[i, j]$ 表示结点 i 与 j 最短路径上的边的权值之和; S 表示所有已被竞争者 A 占有的正则点的集合; S_{nt} 表示所有被竞争者 A 占有的非正则点的集合; $sp[i, j]$ 表示结点 i 与 j 最短路径上的边的集合; $dmin[j]$ 表示正则点 j 与选中结点集 S 中最短路径权值最小的权值 $dmin[j] = \min(spm(i, j)) i \in S$; $dmin_j[j]$ 表示选中结点集 S 的一个结点的编号, 其与顶点 j 的最短路径的权值最小; $power[j]$ 表示竞争者 A 对正则点 j 的竞争力.

2.5 算法的初始状态

在本算法中,共有 m 个初始状态. 即竞争者 A 占有 m 个正则点中的一个.

2.6 竞争力函数、决策函数

a. 竞争力函数

本算法采用一个竞争力函数,其基本思想可描述为:对所有未被选中的正则点,寻找其到 S 中结点的最短路径,最短路径权值越小的结点竞争力越大. 其数学定义为

$$power[j] = \begin{cases} -spm[i, j] & i \in S, j \in (P-S) \\ 0 & j \in S \end{cases}$$

b. 决策函数

只采用一个决策函数,即选中竞争力函数 $power[j]$ 值最大的决策变量. 若两个决策变量的 $power[j]$ 值相同时,则选择编号小的结点.

2.7 资源交换规则

在竞争决策算法中,当竞争进入了一个稳定状态后,可以通过资源交换规则使竞争进入一个非稳定的状态. 对于一个已经找到的 Steiner 生成树,可以通过增加一未选中的 Steiner 点或删除一已选中的 Steiner 点来打破稳定状态,若这种交换能够得到更好的解,则起到了进化的目的.

2.8 算法流程

图的 Steiner 最小树的竞争决策算法流程如下:

算法 CDA_GSTP

输入 n 为顶点个数; m 为正则点个数; $w[i, j]$ 为权值矩阵; V 为顶点集; P 为正则点集; $edges[i]$ 为顶点 i 邻接的所有边集.

输出根据保留最优解的 $arrive$ 矩阵和 w 矩阵得到 Steiner 最小树的路径和总权值.

Step 1 重复剔除整体上的劣质资源或重复固定整体上的优秀资源

调用降阶子程序,按照定理 1~4 删除一定不在 Steiner 最小树上的结点和边资源.

Step 2 初始化

最大的竞争步数为 $m-1$; 初始状态个数 $p = m$; 计算各个节点间的最短路径并保留在 sp 和 spm 矩阵中.

step 3 竞争、决策及资源交换

for $p = 1$ to m

{

$S = \{v_p\}$; // v_p 为第 p 个正则点的编号;

初始化 $arrive$ 为 false;

根据邻接矩阵 $arrive$ 计算 $arrive_t$ 的邻接

情况;

在当前 S 值下,计算每个未选中正则点的竞争力函数值.

Step 3.1 本轮竞争阶段 1:资源分配和争夺阶段

选择竞争力函数最大的顶点编号,记为 dot_1 ,并记 $\text{dot_2} = d\min_j[\text{dot_1}]$.

将结点 dot_2 和 dot_1 的最短路径上的边加入到图中,并将经过的正则点加入到 S 中,非正则点加入到 S_{nt} 中.修改相应结点的 *arrive* 和 *arrive_t* 矩阵的值;

重新计算一部分结点的竞争力函数的值(这里只重新计算因为边 $\text{sp}(\text{dot_1}, \text{dot_2})$ 上的边加入到图中而发生变化的点);

检查 S 中是否已经有 m 个结点,若不是则跳转到 step 3.1 继续执行,否则跳转到 step 3.2;

Step 3.2 本轮竞争阶段 2:资源交换阶段

根据矩阵 *arrive* 和权值矩阵得到 Steiner 最小树的路径和总权值.

执行 addNode 操作;

执行 RemoveNode 操作;

}

Step 4 输出竞争决策得到的结果解

addNode 操作定义如下:

变量 e_1^*, e_2^*, e_3^* 分别用来保存预添加的非正则结点关联的最短边、次短边以及两最短边端点最短路径上的最重边;

$t^* = \infty$;

遍历每一个不在 S_{nt} 中的 Steiner 点 v_s :

a. 找到其关联的最短边 e_1 和次短边 e_2 , 它们的另一端点分别为 p, q ;

b. 根据 $\text{sp}[p, q]$ 找到最短路径上的最重边(即权值最大的边)为 e_3 , 其两个端点分别为 s 和 t ;

c. 记 $t = w(e_1) + w(e_2) - w(e_3)$, 若 $t < 0$, 且 $t < t^*$, 则 $e_1^* = e_1, e_2^* = e_2, e_3^* = e_3, t^* = t$;

若 $t^* \neq \infty$ 时,删除边 e_3^* , 添加边 e_1^* 和 e_2^* .

RemoveNode 操作的定义如下:

遍历每一个在 S_{nt} 中的 Steiner 点 v_s , 其在图中邻接的 k 个结点分别为 $v'_1, v'_2, \dots, v'_k$, 并记这些边的权值之和为 u . 若将 v_s 删除时,则图变成了 k 个不连通的子图.此时分别根据 $v'_1, v'_2, \dots, v'_k$ 的传递闭包集可以得到各个子图的结点集. $V_{T_1}, V_{T_2}, \dots, V_{T_k}$ 分别表示子图的结点集, $E_{T_1}, E_{T_2}, \dots, E_{T_k}$ 分别表示子图的边集, 且有 $V_{T_1} \cap V_{T_2} \cap \dots \cap V_{T_k} = \Phi$,

$V_{T_1} \cup V_{T_2} \cup \dots \cup V_{T_k} = V, E_{T_1} \cap E_{T_2} \cap \dots \cap E_{T_k} = \Phi$.

根据式 $E(i, j) = \{(h, g) | w(h, g) < \infty \text{ 且权值最小, 结点 } h \text{ 在子图 } T_i \text{ 上, } g \text{ 在子图 } T_j \text{ 上, } i \neq j\}$ 用来寻找连通两个子图 T_i 和 T_j 的最短割边.

若 k 个子图中存在 $k-1$ 条最短割边时,计算 $k-1$ 条最短割边的权值,并与 u 比较大小,若小于 u 时,则执行删除操作.

3 数值算例

为验证算法的有效性,采用 Delphi 7.0 在 PC 机上实现了该算法.本算法对大量的算例进行求解,这些测试实例可以通过 <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files> 下载,下面给出几个算例的求解结果.

算例 1 steinb1.txt 中的算例,其中结点个数为 50,正则点个数为 9,边的条数为 63.

首先采用降阶子程序后,排除 $\{1, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 19, 23, 25, 26, 29, 31, 32, 33, 38, 39, 42, 43, 45, 46, 50\}$ 这些结点,共排除 28 个结点.降阶后剩下的边数为 32.执行算法后得到的 Steiner 最小树的权值为 79,其连线为 $\{2-18, 2-21, 12-21, 18-22, 18-28, 20-22, 20-27, 20-48, 21-22, 24-28, 27-34, 28-47, 36-41, 36-49, 37-47, 41-47\}$.

算例 2 steinb7.txt 中的算例,其中结点个数为 75,正则点个数为 13,边的条数为 94.

首先采用降阶子程序后,共排除 39 个结点,结点规模降低一半多,排除的结点为 $\{1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 17, 19, 21, 22, 25, 31, 33, 34, 35, 37, 40, 45, 46, 49, 50, 53, 54, 56, 57, 58, 62, 64, 66, 68, 69, 71, 72, 73, 75\}$.降阶后边的条数为 52,降阶后边的规模也接近降低一半.

执行算法后得到的 Steiner 最小树的权值为 111,其连线为 $\{9-23, 9-26, 9-70, 14-48, 24-63, 26-44, 28-44, 28-48, 28-63, 29-30, 30-42, 36-44, 38-60, 38-65, 42-65, 43-51, 43-55, 43-70, 44-52, 44-65, 59-65\}$.

算例 3 steinc1.txt ~ steinc10.txt 中的 10 个算例,这些算例中结点的个数均为 500.

表 1 为所求得的结果与文献[2]中公布的最优解的比较,表中 n 表示结点的规模, m 表示正则点的规模, $|e|$ 表示边的规模. z^* 是竞争决策算法求得的最优解.

表 1 所求结果与文献[2]最优解的比较
Tab.1 Contrast between z^* and solutions
of reference [2]

算例名称	问题规模 ($n/m/ e $)	降阶后 ($n/m/ e $)	z^*	文献[2] 最优解
C ₁	500/5/625	155/5/274	84	85
C ₂	500/10/625	145/10/256	144	144
C ₃	500/83/625	210/83/331	748	754
C ₄	500/125/625	251/125/375	1 085	1 079
C ₅	500/250/625	337/250/458	1 575	1 579
C ₆	500/5/1 000	372/5/850	55	55
C ₇	500/10/1 000	387/10/874	102	102
C ₈	500/83/1 000	398/83/885	509	509
C ₉	500/125/1 000	424/125/916	718	707
C ₁₀	500/250/1 000	445/250/936	1 103	1 093

从表中可以看出,通过 CDA_GSTP 算法对较大规模的几个算例求得的解,基本上能接近最优解,有的甚至优于公布的最优解.

4 结束语

GSTP 问题是很多问题的原始模型,在工程实践与科学研究中都有着非常重要的意义.但是由于其 NP 困难的特性,限制了精确算法对较大规模问题的求解.竞争决策算法是一种能广泛应用于求解各类组合优化难题的新型寻优算法,其通用性和实用性都比较强.本文在对 GSTP 问题性质分析的基础上,首先对问题进行降阶处理,这可以降低问题求解的难度.然后提出用竞争决策算法的通用流程来求解降阶后的 GSTP 问题,并用 Delphi 实现了算法.为了测试算法的性能,本文求解了 OR-Library 中 78 个算例.实验结果表明,算法求解的速度与效果都比较好.

参考文献:

[1] Karp R M. Reducibility among combinatorial problems, Complexity of computer computations [M]. New York: Plenum Press,1972.

[2] Luyet L,Varone S,Zufferry N. An ant algorithm for the steiner tree problem in graphs[J]. Applications of Evolutionary Computing, Lecture Notes in Computer Science,2007,4448:42 – 51.

[3] Martins S L, Resende M G C, Ribeiro C C, et al. Parallel grasp for the steiner tree problem in graphs using a hybrid local search strategy [J]. Journal of Global Optimization,2000,17(1/2/3/4):267 – 283.

[4] Kapsalis A, Rayward-Smith V J, Smith G D. Solving the graphical steiner tree problem using genetic algorithms [J]. The Journal of the Operational Research Society,1993,44(4):397 – 406.

[5] 宁爱兵,马良,熊小华. 竞争决策算法原理及其应用 [J]. 上海理工大学学报,2008,30(4):369 – 373.

[6] 宁爱兵,马良. 竞争决策算法及其在车辆路径问题中的应用[J]. 管理科学学报,2005,8(6):10 – 18.

[7] 宁爱兵,马良. 度约束最小生成树(DCMST)的竞争决策算法[J]. 系统工程学报,2005,20(6):630 – 634.

[8] 熊小华,马良,宁爱兵. 基于多交换邻域搜索的多维 0/1 背包问题竞争决策算法[J]. 系统工程理论与实践,2010,30 (8):1548 – 1456.

[9] 宁爱兵,马良. 基于快速下界估算的瓶颈旅行商问题竞争决策算法[J]. 上海理工大学学报,2005,27 (3): 223 – 228.

[10] 熊小华,马良,宁爱兵. 多目标最小生成树的竞争决策算法[J]. 系统工程,2010,28 (4):89 – 93.

[11] 宁爱兵,熊小华,马良. 多目标旅行商问题竞争决策算法[J]. 计算机工程与应用,2009,45(34):13 – 16.

[12] 左孝凌,李为监,刘永才. 离散数学[M]. 上海:上海科学技术文献出版社,1982.