

基于加权节点的 Steiner 树启发式算法

赵礼峰, 王小龙*

(南京邮电大学 理学院, 南京 210023)

(* 通信作者电子邮箱 xl365t@126.com)

摘要: Steiner 最小树问题是一个 NP 完全问题, 被广泛应用在通信网络中点到多点的路由选择。为了实现更多链路的共享, 减少所求 Steiner 树的费用, 提出了一种基于加权节点求解 Steiner 树的启发式 (NWMPH) 算法。该算法构造了非正则点的权值公式, 给每一个非正则点赋权值, 根据权值对链路费用进行修正, 通过修正费用最短路径依次把所有的正则点连接起来, 得到包含所有正则点的最小树。对 STEINLIB 标准数据集中的部分数据进行计算, 结果表明: NWMPH 算法与 MPH 算法所用时间基本相同, 得到的 Steiner 树费用优于 MPH 算法; NWMPH 算法比 KBMPH 算法所用时间少, 得到的 Steiner 树费用绝大多数优于 KBMPH 算法。

关键词: MPH 算法; 加权节点; Steiner 树; 启发式算法; 最短路径

中图分类号: TP393.2 **文献标志码:** A

Steiner tree heuristic algorithm based on weighted node

ZHAO Lifeng, WANG Xiaolong*

(School of Natural Sciences, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu 210023, China)

Abstract: Minimum Steiner tree problem is a NP complete problem, and widely used in communication network point to multi-point routing. In order to realize more link sharing, reduce the cost of the desired Steiner tree, an algorithm named NWMPH (Node Weight based Minimum cost Path Heuristic) was proposed to solve the Steiner tree based on weighted node. The algorithm constructed a weighted formula of nonregular points, for each nonregular point weighting value. According to the weights of modifying the link cost. By modifying the cost shortest path in order to connect all regular points, get the minimum tree containing all regular points. For part of the data to calculate STEINLIB standard data set, the results show that: NWMPH algorithm and MPH algorithm used basically the same time. The cost of NWMPH algorithm to get Steiner tree is less than that of MPH algorithm. NWMPH algorithm uses less time and costs less to get Steiner tree than KBMPH algorithm.

Key words: Minimum cost Path Heuristic (MPH) algorithm; weighted node; Steiner tree; heuristic algorithm; shortest path

0 引言

随着通信网络技术的发展, 在许多业务上都需要用到多播。多播性能的好坏主要取决于多播路由算法的好坏, 也就是优化多播树的网络费用问题, 该问题可归化为寻找图上的 Steiner 最小树问题。图的 Steiner 最小树问题是图论中的一个 NP 完全问题^[1]。

解决该问题的精确算法有很多, 如 Hakimi^[2] 提出的枚举法、Dreyfuss 等^[3] 给出的动态规划算法、Beasley^[4] 给出的拉格朗日松弛法等, 虽然这些算法能够得到最优解, 但计算量随着问题规模的增大而呈指数级增长, 不适合求解大规模网络问题。

启发式算法虽然绝大多数情况下不能得到最优 Steiner 树, 但它们的时间复杂度不高, 能够在较短的时间内找到接近最优的准 Steiner 最小树, 因此更具有实际应用价值。比较著名并且性能较好的启发式算法有 KMB (Kou-Mrakovsky-Bermaa) 算法^[5]、MPH (Minimum cost Path Heuristic) 算法^[6]、ADH (Average Distance Heuristic) 算法^[7] 等, 以及基于上述算法的改进算法, 如 KBMPH (Key node Based MPH) 算法^[8]、基

于 KMB 算法的进化规划算法^[9] 等。

本文首先通过分析图中非正则点对构建 Steiner 最小树的影响, 构造权值公式, 计算每一个非正则点的权值。权值越小, 成为 Steiner 点的可能性越大; 反之权值越大, 成为 Steiner 点的可能性越小。在 MPH 算法的基础上, 根据最短路径经过的非正则点的权值求平均权值, 对最短路径费用进行修正, 提出了一种改进的启发式算法——基于加权节点的启发式 (Node Weight based Minimum cost Path Heuristic, NWMPH) 算法。该算法既考虑到了每一个非正则点成为 Steiner 点的可能性, 又实现了更多的链路共享。

1 问题的定义

给定无向连通图 $G = (V, E)$, 其中 V 为顶点集, E 为边集, 定义权值函数 $f: E \rightarrow \mathbf{R}_+$, 构成网络 $N(G, f)$ 。给定子集 $P \subseteq V$, $D = V - P$, 要求在网络 $N(G, f)$ 上寻找一棵子树 $T = (Y, U)$, 使得 $P \subseteq Y \subseteq V$, $U \subseteq E$, 且 $\sum_{e \in U(T)} f(e)$ 最小。称 T 为图 G 关于 P 的 Steiner 最小树。点集 p 中的点称为正则点 (或多播点), 点集 D 中的点称为非正则点 (或非多播点)。点集 S 称为 Steiner-

收稿日期: 2014-07-14; 修回日期: 2014-08-24。

作者简介: 赵礼峰 (1959 -), 男, 安徽淮北人, 教授, 主要研究方向: 图论及其应用、网络流算法; 王小龙 (1989 -), 男, 河南新乡人, 硕士研究生, 主要研究方向: 图论及其应用、网络流算法。

点,简称 s-点,其中 $S \subseteq V \setminus P, Y = P \cup S$ 。记 $|V| = n, |E| = l, |P| = m, r = m - n$ 。

2 求解 Steiner 最小树问题的启发式算法

2.1 MPH 算法

MPH 算法是由 Takahashi 等^[6]在 1980 年提出的,其具体步骤如下:

步骤 1 从正则点集合 P 中任选一个顶点 v_1 , 设置 $i = 1, T_i = \{v_1\}, V_i = \{v_1\}$ 。

步骤 2 对 $i = 2, 3, \dots, m$, 寻找节点 $v_i \in D - V_{i-1}$, 使得 v_i 到 T_{i-1} 的费用最低, 将 v_i 通过费用最短路径 $Path(v_i - T_{i-1})$ 连接到 T_{i-1} 上, $T_i = T_{i-1} \cup Path(v_i - T_{i-1})$, V_i 为 T_i 的节点集, 最终得到的树就是 T_{MPH} 。

2.2 KMB 算法

KMB 算法(又称为 DNH(Distance Network Heuristic)算法)是基于最短路径和最小生成树的算法,其具体算法如下:

步骤 1 构造图 G 的完全距离图 $G' = (V', E')$, 在 G' 中求包含 W 的子图 G_1 。

步骤 2 求 G_1 的最小生成树 T_1 , 然后把 T_1 中的每一条边, 依据 G 中的最短路径扩充为 G 的一个子图 G_2 。

步骤 3 求 G_2 的最小生成树 T_2 , 删除 T_2 中非正则点的叶子节点, 最终得到的树为 T_{KMB} 。

2.3 KBMPH 算法

KBMPH 算法是对 MPH 算法的改进, 在该算法中, 把有较多最短路径经过的节点叫作关键节点(Key Node, KN), 能够在一定条件下优先采用包含了 KN 的路径, 则有可能后面的正则点到树上的最短路径也经过这些 KN, 由此实现更多链路的共享。其具体步骤如下:

步骤 1 求 G 中所有节点对之间的最短路径, 若某条最短路径经过某个节点(这里头尾节点除外), 则将该节点的权值加 1, 然后找到权值最大的 k 个节点, 加入到节点集 F 。

步骤 2 从正则点集合 P 中任选一个顶点 v_1 , 设置 $i = 1, T_i = \{v_1\}, V_i = \{v_1\}$ 。

步骤 3 对 $i = 2, 3, \dots, m$, 寻找节点 $v_i \in D - V_{i-1}$, 使得 v_i 到 T_{i-1} 的修正费用 $d(v_i - T_{i-1})$ 最低, 若路径经过一个以上 F 集中的节点, 则将路径的费用乘以系数 λ 后再作比较, $0 < \lambda < 1$ 。将 v_i 通过修正费用最短路径 $Path(v_i - T_{i-1})$ 连接到 T_{i-1} 上, $T_i = T_{i-1} \cup Path(v_i - T_{i-1})$, V_i 为 T_i 的节点集, 最终得到的树就是 T_{KBMPH} 。

2.4 算法的比较

从算法的时间复杂度上考虑, MPH 算法的复杂度为 $O(mn^2)$, KMB 算法的复杂度为 $O(mn^2)$, KBMPH 算法的复杂度为 $O(n^3)$, 由于 $m < n$, 所以 MPH 算法和 KMB 算法在时间复杂度方面优于 KBMPH 算法。从性能方面考虑, MPH 算法的准 Steiner 树的树长要比 KMB 算法的小, 在大多数情况下, KBMPH 算法的准 Steiner 树的树长要比 KMB 算法的小。

3 改进的算法

KBMPH 算法虽然在性能方面进行了改进, 但是增加了算法的时间复杂度。由于 KBMPH 算法是选取固定的 k 个关键节点, 对路径的费用进行修正, 而这 k 个关键节点不一定是

最后的 Steiner 点, 并且其余的非正则点也有可能成为 Steiner 点。考虑到这种可能性, 本文构造了一种对所有非正则点进行权值计算的加权公式, 并提出了一种基于加权节点的启发式算法 NWMPH 算法。每个非正则点的权值与该点属于 Steiner 点的概率成反比, 权值越大属于 Steiner 点的可能性就越小, 否则相反。

通过对网络进行分析, 我们得出影响权值的因素主要有^[10-11]: 第一, 正则点最短路径经过的次数。非正则点被越多的最短路径经过, 越有可能通过费用最低路径连接到树上。第二, 节点的度数。度数越大, 说明该节点连接的顶点越多, 其在图中占有重要的位置, 更有可能被其他路径经过。第三, 直接与正则点连接的次数。直接与正则点连接的次数越多, 成为 Steiner 点的可能性就越大, 起到连接正则点之间桥梁的作用。其中正则点最短路径经过次数的影响因素最大。

3.1 变量设置及公式

d_i 表示节点 i 的度数; g_i 表示任意正则点对最短路径经过节点 i 的次数; l_i 表示节点 i 直接与正则点相连接的次数; λ 表示设最短路径经过 s 个非正则点, 对这 s 个非正则点的权值求的平均权值; $d(v_i, T_{i-1})$ 表示 v_i 到树 T_{i-1} 的最短距离; $d'(v_i - T_{i-1})$ 表示 v_i 到树 T_{i-1} 的修正费用。

$$d'(v_i, T_{i-1}) = \lambda \times d(v_i, T_{i-1}) \quad (1)$$

由于在正则点最短路径上的点成为 Steiner 点的可能性较大, 因此本文通过变量 α 来控制其重要程度:

$$f(i) = \alpha g_i + \frac{1}{2}(1 - \alpha)(d_i + l_i) \quad (2)$$

其中 $0.4 \leq \alpha \leq 0.8$ 。

令 F 为 $f(i)$ 的最大值, 则节点 i 的权值为:

$$W(i) = \exp(-\beta \times f(i)/F) \quad (3)$$

其中 $0.1 \leq \beta \leq 2$ 。

当 $\beta = 0.1$ 时 $W(i)$ 的最小值为 0.904 84; 当 $\beta = 2$ 时 $W(i)$ 的最小值为 0.135 34。

3.2 算法过程描述

步骤 1 求 G 中所有正则点对之间的最短路径, 对每一个非正则点, 统计正则点对之间最短路径经过的次数。

步骤 2 对 G 中的所有非正则点按式(3)计算权值。

步骤 3 从点集 D 中任选一个顶点 v_1 , 初始生成 $T_1 = \{v_1\}, V_1 = \{v_1\}$ 。

步骤 4 对 $i = 2, 3, \dots, m$, 寻找节点 $v_i \in D - V_{i-1}$, 使得 v_i 到 T_{i-1} 的修正费用 $d'(v_i - T_{i-1})$ 最低, 然后将 v_i 通过修正费用最短路径 $Path(v_i - T_{i-1})$ 连接到 T_{i-1} 上, $T_i = T_{i-1} \cup Path(v_i - T_{i-1})$, V_i 为 T_i 的节点集, 最终得到的树就是 T_{NWMPH} 。

3.3 算法的正确性和时间复杂度

定理 1 NWMPH 算法是覆盖全部正则点的无环的准 Steiner 树。

证明 1) 由于算法的步骤 4, 是逐步添加正则点到树上, 直到正则点集为空, 因此最后形成的树肯定包含了所有的正则点。

2) 下面证明是无环的。正则点与树相连接的最短路径是利用 Dijkstra 算法的结果, 这条路径与树不相连的部分是不会形成环路的。假设与树相连的部分形成环路, 则必有两个以上的节点与树相连接, 那么该正则点到其中一个节点的

路径才是正则点到已有树的最短路径,所以假设不成立,该路径与树只有一个节点相连,不会形成环路。由于修正费用最小的路径是对上述最短路径的费用进行修正,并没有改变最短路径,所以修正费用最小的路径与树也不会构成环路。

定理 2 NWMPH 算法的时间复杂度为 $O(mn^2)$ 。

证明 步骤 1 中对 m 个正则点利用 Dijkstra 算法求出正则点对之间的最短路径,其时间复杂度是 $O(mn^2)$ 。

步骤 2 中对每条路径上的节点分别计算权值,其时间复杂度为 $O(n)$,共需要对 $m(m-1)$ 条路径运算,所以其时间复杂度为 $O(m^2n)$ 。

步骤 3 和步骤 4 需要将 m 个正则点分别通过修正费用最小的路径连上,对每个正则点选择修正最小费用路径时最多对 $m-1$ 条路径进行比较,所以其时间复杂度为 $O(m^2)$ 。由于 $m < n$,所以算法的总复杂度为 $O(mn^2)$ 。

4 实例分析

实例 1 求解图 1 的 Steiner 最小树。

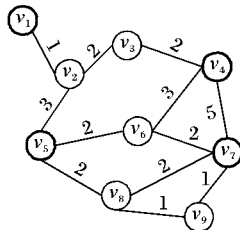


图 1 求 Steiner 最小树的原图

1) KMB 算法和 MPH 算法得到树的最小费用都是 12。

2) KBMPH 算法:如果随机选择的节点为 v_1 , 则可以得到最优解,树的最小费用为 11。但是当随机选择的节点为 v_5 , 由于 v_2, v_6, v_8 都是关键节点,并且修正费用的系数 λ 固定,有些时候得到的不是最优解,如图 2 所示,求得树的费用为 12。

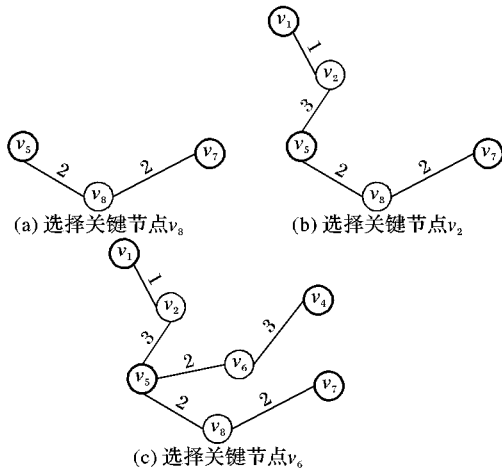


图 2 KBMPH 算法的示例

3) NWMPH 算法。

步骤 1 $g_2 = 2, g_3 = 1, g_6 = 3, g_8 = 1, g_9 = 1$ 。

步骤 2 令 $\alpha = 0.6, \beta = 0.5$, 求得 $W(2) = 0.6271, W(3) = 0.8187, W(6) = 0.6065, W(8) = 0.7659, W(9) = 0.8187$ 。由于权值越小,对路径费用的修正越大,得到的修正费用越小,因此选择非正则节点的先后顺序为 v_6, v_2, v_8, v_3, v_9 。

步骤 3 任选一项点例如 v_5 , 其他正则点到 v_5 的最短路径费用为 4, 这样的路径有 4 条,通过计算修正费用得到 $v_7-v_6-v_5$

的修正费用最小,将 v_7 通过 v_6 添加到树上。同理,依次添加 v_1, v_4 节点到树中,得到树的最小费用为 11,如图 3 所示。

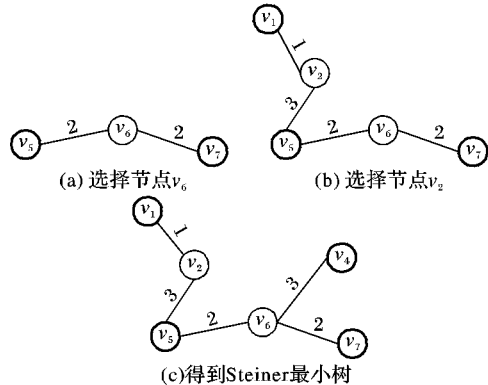


图 3 NWMPH 算法的示例

实例 2 用 STEINLIB 标准数据集中^[12]的部分数据进行实验,求出用 MPH、KBMPH、NWMPH 三种算法所得到的 Steiner 树费用,如表 1 所示,其中 OptSol(Optimal Solution) 表示测试用例的最优解。对测试用例用三种算法分别运行 15 次,得到平均的运行时间,如图 4 所示。

表 1 三种算法所得 Steiner 树费用

测试用例	图的规模 $n/L/m$	OptSol	Steiner 树费用		
			MPH	KBMPH	NWMPH
B01	50/63/9	82	82	82	82
B02	50/63/13	83	84	83	83
B03	50/63/25	138	138	138	138
B04	50/100/9	59	59	59	59
B05	50/100/13	61	61	61	61
B06	50/100/25	122	124	124	124
B07	75/94/13	111	111	111	111
C01	500/625/5	85	86	86	85
C02	500/625/10	144	146	144	144
C04	500/625/125	1 079	1 184	1 086	1 094
D03	1 000/1 250/167	1 565	1 623	1 625	1 594
D05	1 000/1 250/500	3 250	3 326	3 283	3 309

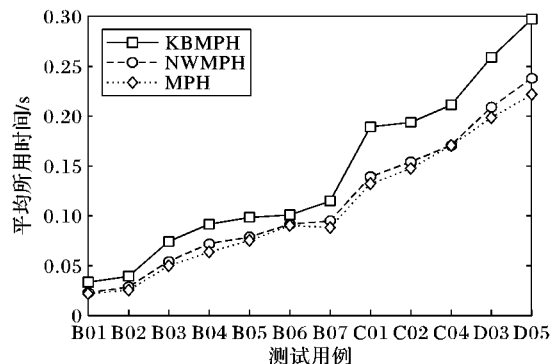


图 4 三种算法平均所用时间比较

由表 1 的测试结果对比可以发现, NWMPH 算法得到的 Steiner 树费用优于 MPH 算法, 绝大多数优于 KBMPH 算法。只有当测试用例中正则点数目较大时, NWMPH 算法得到的 Steiner 树费用大于 KBMPH 算法。图 4 的实验结果表明 NWMPH 算法与 MPH 算法所用时间基本一致, KBMPH 算法所用的时间最长。这与我们之前所分析的时间复杂度吻合。

(下转第 3457 页)

偏好信息的演化规律都是值得进一步研究的问题。

参考文献:

- [1] LIU J, ZHOU T, WANG B. Research progress of personalized recommendation systems[J]. Progress in Natural Science, 2009, 19(1): 1–15. (刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 19(1): 1–15.)
- [2] CHEN Y, CHENG L, CHUANG C. A group recommendation system with consideration of interactions among group members[J]. Expert Systems with Applications, 2008, 34(3): 2082–2090.
- [3] DIAZ AGUDO B, WATSON I. A case-based solution to the cold-start problem in group recommenders[C]// Proceedings of the 2012 International Conference on Case-based Reasoning, LNCS 7466. Berlin: Springer, 2012: 342–356.
- [4] CARVALHO L A M C, MACEDO H T. Users' satisfaction in recommendation systems for groups: an approach based on noncooperative games[C]// Proceedings of the 22nd International Conference on World Wide Web Companion. New York: ACM, 2013: 951–958.
- [5] ANAND D. Group movie recommendations via content based feature preferences[J]. International Journal of Scientific and Engineering Research, 2013, 4(2): 1–5.
- [6] BAATARJAV E A, PHITHAKKITNUKON S, DANTU R. Group recommendation system for facebook[C]// On the Move to Meaningful Internet Systems: OTM 2008 Workshops, LNCS 5333. Berlin: Springer, 2008: 211–219.
- [7] QUIJANO-SANCHEZ L, BRIDGE D, DIAZ-AGUDO B, *et al.* Case-based aggregation of preferences for group recommenders[M]. Case-Based Reasoning Research and Development. Berlin: Springer, 2012: 327–341.
- [8] GARCIA I, SEBASTIA L, PAJARES S, *et al.* Approaches to preference elicitation for group recommendation[C]// Proceedings of 2011 International Conference on Computational Science and Applications. Berlin: Springer, 2011: 20–23.
- [9] MATEOS A, JIMENEZ A. A trapezoidal fuzzy numbers-based approach for aggregating group preferences and ranking decision alternatives in MCDM[C]// Proceedings of 5th International Conference on Evolutionary Multi-Criterion Optimization, LNCS 5467. Berlin: Springer, 2009: 7–10.
- [10] GARCIA I, PAJARES S, SEBASTIA L, *et al.* Preference elicitation techniques for group recommender systems[C]// ICCSA 2011: Computational Science and Its Applications, LNCS 6786. Berlin: Springer, 2012, 189: 155–175.
- [11] LIU P. A weighted aggregation operators multi-attribute group decision-making method based on interval-valued trapezoidal fuzzy numbers[J]. Expert Systems with Applications, 2011, 38(1): 1053–1060.
- [12] CHEN S-H. Fuzzy numbers with maximizing set and minimizing set[J]. Fuzzy Sets and Systems, 1985, 17(2): 113–129.
- [13] CHEN S-J, CHEN S-M. A new method for handling multi criteria fuzzy decision-making problems using FN-IOWA operators[J]. Cybernetics and Systems, 2003, 34(2): 109–137.
- [14] CHENG C. A new approach for ranking fuzzy numbers by distance method[J]. Fuzzy Sets and Systems, 1998, 95(3): 307–317.
- [15] MOLODTSOV D. Soft set theory-first results[J]. Computers and Mathematics with Applications, 1999, 37(4): 19–31.
- [16] ROY A R, MAJI P K. A fuzzy soft set theoretic approach to decision making problems[J]. Journal of Computational and Applied Mathematics, 2007, 203(2): 412–418.
- [17] YANG X, LIN T, YANG J, *et al.* Combination of interval-valued fuzzy set and soft set[J]. Computers and Mathematics with Applications, 2009, 58(3): 521–527.

(上接第3416页)

5 结语

本文在 MPH 算法的基础上,通过对每一个非正则点赋权值,并根据权值对最短路径进行修正,增加了链路的共享。提出的 NWMPH 算法时间复杂度与 MPH 算法相同,比 KBMPH 算法低。实例 1 说明了 NWMPH 算法比 KBMPH 算法的稳定性高,实例 2 的实验结果表明,NWMPH 算法所求解优于 MPH 算法,并且在绝大多数情况下优于 KBMPH 算法。因此,NWMPH 算法是一个时间复杂度较低、性能较好的算法。

参考文献:

- [1] KARP R M. Complexity of computer computations[J]. Reducibility Among Combinatorial Problems, 1972, 23(1): 85–103.
- [2] HAKIMI S L. Steiner's problem in graphs and its implications[J]. Networks, 1971, 1(2): 113–133.
- [3] DREYFUS S E, WAGNER R A. The Steiner problem in graphs[J]. Networks, 1971, 1(3): 195–207.
- [4] BEASLEY J E. An SST-based algorithm for the Steiner problem in graphs[J]. Networks, 1989, 19(1): 1–16.
- [5] KOU L, MARKOWSKY G, BERMAN L. A fast algorithm for Steiner trees[J]. Acta Informatica, 1981, 15(2): 141–145.
- [6] TAKAHASHI H, MATSUYAMA A. An approximate solution for the Steiner problem in graphs[J]. Math Japonica, 1980, 24(6): 573–577.
- [7] RAYWARD-SMITH V J, CLARE A. On finding Steiner vertices[J]. Networks, 1986, 16(3): 283–294.
- [8] YU Y, QIU P. An improved Steiner tree heuristic algorithm[J]. Journal of communications, 2002, 23(11): 35–40. (余燕平, 仇佩亮. 一种改进的 Steiner 树启发式算法[J]. 通信学报, 2002, 23(11): 35–40.)
- [9] GUO W, XI Y, QUAN Y. Heuristic evolutionary programming to solve the Steiner tree problem[J]. Journal of Shanghai Jiao Tong University, 2001, 35(8): 1152–1154. (郭伟, 席裕庚, 全亚斌. 启发式进化规划求解 Steiner 树问题[J]. 上海交通大学学报, 2001, 35(8): 1152–1154.)
- [10] YANG N, HU Y. Steiner tree heuristic algorithm based on weight[C]// Proceedings of the 2nd International Conference on Future Computer and Communication. Piscataway: IEEE, 2010, 3: 415–418.
- [11] YANG N. Application layer multicast and Steiner algorithm research[D]. Dalian: Dalian University of Technology, 2010. (杨宁. 应用层多播与 Steiner 算法的研究[D]. 大连: 大连理工大学, 2010.)
- [12] BEASLEY J E. OR-Library: distributing test problems by electronic mail[J]. Journal of the Operational Research Society, 1990, 41(11): 1069–1072.