

多目标 MIN-MAX 度最小树问题及其求解

魏 欣, 马 良

(上海理工大学 管理学院, 上海 200093)

摘要: 在多目标最小生成树问题和 MIN-MAX 度最小树问题的基础上, 探讨使生成树最大顶点度数以及总权重都尽可能小的另类多目标 MIN-MAX 度最小生成树问题。分析了这一特殊的顶点度约束与 Hamilton 路的关联性质, 在此基础上设计了先 Hamilton 路再 MIN-MAX 度最小树的独特求解方案。根据初始条件不同, 当网络图不存在 Hamilton 路时, 引入改进的蚁群优化算法, 将转移概率由基本的指数形式改进为线性形式, 在不影响求解质量的前提下, 提高计算效率。针对以上策略, 设计了相应的求解方案, 并在计算机上用 Delphi 编程实现。大量数值算例验证表明, 算法能快速有效地求解多目标情形下的 MIN-MAX 度最小生成树问题。

关键词: 多目标; MIN-MAX 度; 生成树; Hamilton 路

中图分类号: O 224 **文献标志码:** A

Multi-criteria MIN-MAX Degree Minimum Spanning Tree Problem and Its Solution

WEI Xin, MA Liang

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Based on the multi-criteria minimum spanning tree(MST) problem and the MIN-MAX degree minimum spanning tree problem, a special multi-criteria MIN-MAX degree MST problem was discussed, aiming to minimize both the maximum degree and the total weights of a spanning tree. A unique solution scheme was designed, trying to find the Hamilton path firstly and seeking the MIN-MAX degree MST afterwards, after analyzing the correlation between this particular vertex degree constraint and the Hamilton path. Depending on different initial conditions, an improved ant colony optimization algorithm was introduced to change the transition probability from the basic exponential to linear form if there was no Hamilton path in the network, so as to enhance the computing efficiency without affecting the solution quality. A corresponding solution scheme was designed and coded by Delphi, in view of the strategies proposed above. A number of numerical examples were tested, and the results show that the algorithm proposed is efficient and effective for multi-criteria MIN-MAX degree minimum spanning tree problems.

收稿日期: 2018-03-08

基金项目: 国家自然科学基金资助项目(71401106); 教育部人文社科规划基金资助项目(16YJA630037); 上海市软科学研究重点项目(18692110500)

第一作者: 魏 欣(1984-), 女, 实验师. 研究方向: 系统工程、智能优化. E-mail: usstwx@126.com

通信作者: 马 良(1964-), 男, 教授. 研究方向: 系统工程、智能优化. E-mail: maliang@usst.edu.cn

Keywords: multi-criteria; MIN-MAX degree; spanning tree; Hamilton path

最小生成树(minimum spanning tree, MST)问题是运筹学、网络优化中的一个基本而又经典的优化问题^[1],早在 20 世纪 50 年代就已圆满解决,在生活中有着大量的实际应用。此后,在此基础上,又逐步延伸出一系列的扩展问题,如:度约束最小生成树问题、多目标最小树问题、MIN-MAX 度生成树问题、MIN-MAX 度最小树问题等^[2-8]。由于这些扩展问题在求解难度上与经典的最小树问题截然不同,目前都已归入所谓的 NP 难题范畴。其中,对生成树中各节点的度数有某种要求或限制,以及网络图本身就带有多个权重属性的多目标问题,在现实社会经济生活中经常会遇到,具有重要的应用价值。例如:集成电路布线中往往对相关节点连出的线数有某种工艺限制^[9-10],这就是典型的度约束最小树问题。再如:通信网络实时响应时,为防止某些节点故障带来的全网脆弱性,需使其最大顶点度数尽可能小,这就是典型的 MIN-MAX 度生成树问题。类似地,管道排放系统或紧急疏散过程中,为防止由于某些节点过度拥堵而造成疏通能力下降,也需要使网络中的最大顶点度数尽可能小。此外,在一些需要实时、快速响应的动态网络系统中,更会涉及到带有多个目标的复合问题。

目前,国内外相关文献或仅单独研究度限制 MST 问题,或只探讨多目标意义下的 MST 问题,鲜有文献涉及将两者进行综合考虑。鉴于许多应用都可抽象归结为此类 NP 难题或其混合型问题,而目前这方面的研究又相当欠缺,因此,本文对一般的多目标 MIN-MAX 度最小树问题建立相应的数学模型,并设计一种基于蚁群寻优思想的智能算法。

1 数学模型

设 $G=(V,E,W)$ 为赋权无向连通图(G 中无环,也无多重边),其中, $V=\{1,2,\cdots,n\}$ 为顶点集, n 为顶点个数, $E=\{e_1,e_2,\cdots,e_m\}$ 为边集,各边上权重属性值记为 $w_{ij}^r(w_{ij}^r>0, w_{ii}^r=+\infty, i,j\in V, r=1,2,\cdots,L)$, L 为目标个数,并设:

$$x_{ij}=\begin{cases} 1, & \text{边}(i,j)\text{在最优树上} \\ 0, & \text{其他} \end{cases}$$

多目标 MIN-MAX 度最小树问题可表示为如下数学规划模型:

$$\min Z_0=\max \sum_{j=1}^n x_{ij} \tag{1}$$

$$\min Z_1=\sum_{i=1}^n \sum_{j=1}^n w_{ij}^1 x_{ij} \tag{2}$$

⋮

$$\min Z_L=\sum_{i=1}^n \sum_{j=1}^n w_{ij}^L x_{ij} \tag{3}$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^n \sum_{j=1}^n x_{ij}=n-1 \end{cases} \tag{4}$$

$$\begin{cases} \sum_{i\in S} \sum_{j\in S} x_{ij}\leq |S|-1, \forall S\subset V, S\neq \varnothing \end{cases} \tag{5}$$

$$\begin{cases} x_{ij}\in \{0,1\} \end{cases} \tag{6}$$

式中:目标(1)的作用在于使生成树的最大顶点度数最小;目标(2)~(3)使生成树的 L 个目标函数值都尽可能小,即多目标意义下的 Pareto 有效解;约束(4)则保证了所得解的边数满足树的基本要求 and 性质;约束(5)中 $|S|$ 为集合 S 中所含图 G 的顶点个数,保证了所得解中没有任何子回路的产生。

这里,关于多目标优化的生成树 Pareto 有效解定义如下:

假设一棵生成树 T 是一个 Pareto 解,则意味着不存在任何其他生成树 T' ,使得

$$Z_r(T')\leq Z_r(T), r=1,2,\cdots,L$$

其中,至少有一个不等式严格成立(Z_r 为相应生成树的目标函数值)。

2 算法设计

由于算法中涉及度约束最小树的一个子算法,因此,这里先引入相关概念与思想。

对于 n 个顶点的赋权无向连通图 G ,假定各顶点的度限制为 $b_i(i=1,2,\cdots,n)$,则当各 b_i 为 $n-1$ 时,问题就归结为无度约束情况下的一般最小树问题;当 $2<b_i<n-1$ 时,为度约束最小树问题;而当各 $b_i=2$ 这种特殊情形时,则问题恰好为一个蕴

含 Hamilton 圈的旅行商问题^[8]。

于是,可以采用如下的想法作为整个算法的一个子算法:若给定的网络图存在最小 Hamilton 圈,可先求出相应的最小 Hamilton 路,则对应生成树的最大顶点度恰好为 2,即转化为最大顶点度数最小的 MIN-MAX 度生成树问题。

目前,一般情况下图的 Hamilton 性的充要条件尚未完全解决,理想的方法迄今仍未找到,相关研究仍是图论中一个重要的难题^[11-12]。比较经典的判别定理是:若图 G 中任意两个不相邻顶点 u 和 v 都有 $d(u) + d(v) \geq n - 1$, 则 G 含有一条 Hamilton 路。

由于一般情况下不能保证给定图一定存在 Hamilton 路,因此,本文求解多目标 MIN-MAX 度最小树问题的算法基本思路是:先采用快速启发式算法,寻找图中的最小 Hamilton 路(即最大度为 2 的生成树);若不存在 Hamilton 路,则退而求其次,设计一种改进的蚁群优化算法进行搜索,由单个蚂蚁按转移概率逐个节点进行搜索,则每个蚂蚁经过的节点度数除出发点之外都为 2,并行 n 个蚂蚁后,最终输出的结果将是使得顶点度较小且总权值也相对较小的解。

下面,给出求解多目标 MIN-MAX 度最小树问题的算法主要步骤:

Step 1 读入数据,随机产生 L 个权重 $\lambda_r (r = 1, 2, \dots, L)$, $\sum_{r=1}^L \lambda_r = 1$, 首先考虑加权后各边权重,即 $w_{ij} = \sum_{r=1}^L \lambda_r w_{ij}^r (i, j = 1, \dots, n)$;

Step 2 初始化路径数组 $cycle$ 为 0, 初始点遍历, 设置初始出发点 s , $nearest \leftarrow s$, $i \leftarrow 1$;

Step 3 找寻距离 $nearest$ 最近的点, 置于 $cycle[nearest]$ 中;

Step 4 $i \leftarrow i + 1$, 若 $i < n$, 则转 Step 3;

Step 5 若 $s < n$, 则转 Step 2;

Step 6 将 $cycle[nearest]$ 中最近距离点序列置于路径 $route$ 中, 若 $route$ 中含 0, 则最小 Hamilton 路搜索失败, 转 Step 8, 否则, 转 Step 7;

Step 7 输出最大度为 2 的生成树, 并计算相应 L 个目标值, 即

$$Z_r = \sum_{i,j \in E_T} x_{ij} w_{ij}^r, \quad r = 1, 2, \dots, L$$

Step 8 设置蚁群算法预定迭代次数、轨迹强度常数 Q 、轨迹强度重要性 α 、能见度重要性 β ,

并置当前迭代次数 $count = 0$;

Step 9 将各蚂蚁初始出发点置于当前解集中, 对每个蚂蚁 k , 按概率 $P_{ij}^k(t)$ 转移到顶点 j , 并将顶点 j 置于当前解集, 更新当前各顶点度数;

Step 10 计算各蚂蚁的目标函数值 $Z_k (k = 1, 2, \dots, m)$, 记录当前最小值 Z_{opt} 及相应最小生成树 $T = \{(i, j) \in E_T\}$;

Step 11 对各路径计算信息素轨迹增量 $\Delta\tau_{ij} = \Delta\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k$, 并按轨迹更新方程修改信息素轨迹强度 $\tau_{ij} = \rho\tau_{ij} + \Delta\tau_{ij}$;

Step 12 置 $\Delta\tau_{ij} = 0$, $count = count + 1$, 若 $count$ 小于预定迭代次数且无退化行为, 则转 Step 9, 否则, 转 Step 13;

Step 13 计算当前生成树的 L 个目标函数值并输出 $Z_r = \sum_{i,j \in E_T} x_{ij} w_{ij}^r$, $r = 1, 2, \dots, L$, 及相应生成树 $T = \{(i, j) \in E_T\}$ 。

上述步骤中, Step 2~7 融入了最小 Hamilton 路的求解策略, 使生成树的最大顶点度最小; 当 Hamilton 路不存在时, 则转而进行 Step 8~13 蚁群优化算法的操作。此外, Step 9 中将转移概率由基本的指数形式改进为线性形式, 可大幅提高计算效率, 但同时不影响计算质量, 具体形式可表示为

$$P_{ij}^k = \frac{\alpha\tau_{ij} + \beta\mu_{ij}}{\sum_l [\alpha\tau_{il} + \beta\mu_{il}]}$$

式中: p_{ij}^k 为蚂蚁 k 的转移概率; τ_{ij} 为弧 (i, j) 的轨迹强度; μ_{ij} 为弧 (i, j) 的能见度; α 为轨迹的相对重要性; β 为能见度相对重要性; l 为所有尚未被访问的顶点集。

此外, Step 11 中的信息素浓度增量 $\Delta\tau_{ij}$ 采用 Ant-Density 模型:

$$\Delta\tau_{ij}^k = \begin{cases} Q, & \text{若 } (i, j) \text{ 在最优路径上} \\ 0, & \text{其他} \end{cases}$$

由算法流程可知, 若给定图存在 Hamilton 路, 则算法运行至 Step 7 即可结束, 可省却其后若干步寻优过程, 此时时间复杂度为 $O(n^2)$; 否则, 算法的时间复杂度由蚁群算法决定, 这时, 整个算法的时间复杂度为 $O(count \cdot n^3)$ 。

因此, 相较于其他启发式算法, 本文算法可根据实际图的不同特征, 采取相应优化策略: 当给定图存在 Hamilton 路时, 可大大节省运算时间; 否则, 将采用退而求其次的改进蚁群算法实

施优化计算。

3 计算实验

上述算法用 Embarcadero Delphi 实现，在 Windows 7(64 位)系统下运行通过。为方便起见，仅对双目标问题进行了大量测试，均能在短时间内获得理想效果。下面，给出部分实例计算结果并予以说明。

例 1（源自文献[5]） $n=9$ ， $L=2$ ，权重属性对应邻接矩阵分别为：

$$W^1 = \begin{bmatrix} \infty & 10 & 79 & 41 & 39 & 54 & 31 & 46 & 53 \\ 10 & \infty & 31 & 11 & 30 & 6 & 89 & 9 & 26 \\ 79 & 31 & \infty & 86 & 5 & 84 & 84 & 26 & 75 \\ 41 & 11 & 86 & \infty & 47 & 46 & 1 & 17 & 96 \\ 39 & 30 & 5 & 47 & \infty & 28 & 94 & 71 & 2 \\ 54 & 6 & 84 & 46 & 28 & \infty & 78 & 42 & 55 \\ 31 & 89 & 84 & 1 & 94 & 78 & \infty & 3 & 41 \\ 46 & 9 & 26 & 17 & 71 & 42 & 3 & \infty & 91 \\ 53 & 26 & 75 & 96 & 2 & 55 & 41 & 91 & \infty \end{bmatrix}$$

$$W^2 = \begin{bmatrix} \infty & 62 & 76 & 95 & 50 & 62 & 10 & 13 & 37 \\ 62 & \infty & 89 & 77 & 65 & 71 & 70 & 67 & 17 \\ 76 & 89 & \infty & 100 & 95 & 23 & 10 & 92 & 79 \\ 95 & 77 & 100 & \infty & 7 & 12 & 11 & 84 & 23 \\ 50 & 65 & 95 & 7 & \infty & 89 & 39 & 39 & 38 \\ 62 & 71 & 23 & 12 & 89 & \infty & 84 & 93 & 62 \\ 10 & 70 & 10 & 11 & 39 & 84 & \infty & 33 & 88 \\ 13 & 67 & 92 & 84 & 39 & 93 & 33 & \infty & 68 \\ 37 & 17 & 79 & 23 & 38 & 62 & 88 & 68 & \infty \end{bmatrix}$$

在无顶点度约束的情况下，文献[5]共求得该算例的 28 个 Pareto 解，最大顶点度数为 3 和 4，且集中于某几个关键点。

如同时考虑生成树的最大顶点度尽可能小，则本文算法运行 10 轮后，剔除劣解，可获得的 Pareto 解集如表 1 所示。

表 1 例 1 计算结果

Tab.1 Calculation results of example 1

顶点最大度 d_i	双目标总权重	生成树各边
2	(98, 463)	1-7, 2-1, 3-5, 4-8, 5-9, 6-2, 7-4, 8-3
2	(99, 414)	1-2, 3-8, 4-6, 5-3, 6-2, 7-4, 8-7, 9-5
2	(135, 290)	1-8, 2-9, 4-6, 5-3, 6-2, 7-4, 8-7, 9-5
2	(139, 281)	1-8, 2-1, 4-6, 5-9, 5-3, 7-4, 8-7, 9-2
2	(166, 263)	1-8, 2-9, 4-7, 5-3, 6-4, 7-1, 8-2, 9-5
2	(243, 190)	1-7, 2-6, 4-5, 5-9, 6-3, 7-4, 8-1, 9-2
2	(366, 130)	1-8, 2-9, 3-7, 4-6, 5-4, 6-3, 7-1, 9-5

例 2（源自文献[13]） $n=7$ ， $L=2$ ，图及其权

重值如图 1 所示。

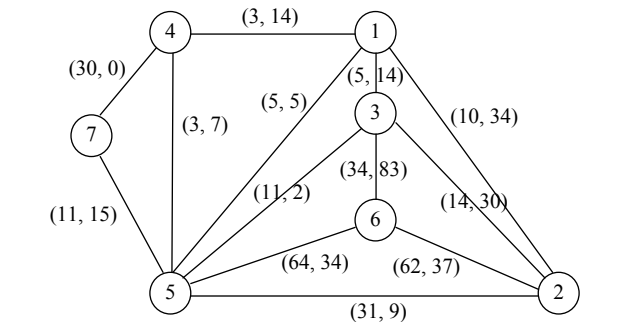


图 1 实例 2 网络图
Fig.1 Network graph of example 2

在无顶点度约束情况下，文献[13]中共求得该算例的 21 个 Pareto 解(原文列出了 22 个，但其中有一个是劣解，应舍去)，其最大顶点度数分别为：3，4，5，6。生成树最大顶点度数越高，稳定性越差。

因此，为使生成树的最大顶点度数尽可能小，运行本文算法 10 轮后，剔除劣解，可得最大顶点度仅为 2 的 Pareto 解如表 2 所示。

表 2 例 2 计算结果

Tab.2 Calculation results of example 2

顶点最大度 d_i	双目标总权重	生成树各边
2	(98, 117)	1-3, 1-4, 2-3, 2-6, 4-5, 5-7
2	(119, 93)	1-3, 1-5, 2-3, 2-6, 4-5, 4-7

例 3 考虑 Hamilton 路不存在的情况， $n=8$ ， $L=2$ ，相关权重值为[1, 50]之间的伪随机数，不存在的边权重值定义为 0，邻接矩阵如下：

$$W^1 = \begin{bmatrix} \infty & 41 & 24 & 19 & 28 & 9 & 28 & 0 \\ 41 & \infty & 27 & 31 & 31 & 0 & 0 & 17 \\ 24 & 27 & \infty & 23 & 13 & 0 & 0 & 22 \\ 19 & 31 & 23 & \infty & 35 & 0 & 0 & 47 \\ 28 & 31 & 13 & 35 & \infty & 0 & 0 & 21 \\ 9 & 0 & 0 & 0 & 0 & \infty & 0 & 0 \\ 28 & 0 & 0 & 0 & 0 & 0 & \infty & 0 \\ 0 & 17 & 22 & 47 & 21 & 0 & 0 & \infty \end{bmatrix}$$

$$W^2 = \begin{bmatrix} \infty & 21 & 10 & 1 & 48 & 25 & 25 & 0 \\ 21 & \infty & 49 & 21 & 6 & 0 & 0 & 8 \\ 10 & 49 & \infty & 28 & 3 & 0 & 0 & 42 \\ 1 & 21 & 28 & \infty & 40 & 0 & 0 & 15 \\ 48 & 6 & 3 & 40 & \infty & 0 & 0 & 26 \\ 25 & 0 & 0 & 0 & 0 & \infty & 0 & 0 \\ 25 & 0 & 0 & 0 & 0 & 0 & \infty & 0 \\ 0 & 8 & 42 & 15 & 26 & 0 & 0 & \infty \end{bmatrix}$$

由于给定图的 Hamilton 路不存在，算法跳过了第一个子算法，直接进入改进的蚁群算法进行

求解, 相关参数按文献[8]的方式设定: $\alpha, \beta = 1, 2, 3$, $\rho = 0.7$, 运行 10 轮后, 剔除劣解, 获得 Pareto 解如表 3 所示。

表 3 例 3 计算结果
Tab.3 Calculation results of example 3

顶点最大度 d_i	双目标总权重	生成树各边
3	(98, 463)	1-7, 2-1, 3-5, 4-8, 5-9, 6-2, 7-4, 8-3
3	(99, 414)	1-2, 3-8, 4-6, 5-3, 6-2, 7-4, 8-7, 9-5
3	(135, 290)	1-8, 2-9, 4-6, 5-3, 6-2, 7-4, 8-7, 9-5
3	(139, 281)	1-8, 2-1, 4-6, 5-9, 5-3, 7-4, 8-7, 9-2
3	(366, 130)	1-8, 2-9, 3-7, 4-6, 5-4, 6-3, 7-1, 9-5

目前, 此类使最大顶点度数最小的多目标 MIN-MAX 度最小树问题的优化研究较少, 相应的求解算法也较为鲜见。本文针对这一问题设计了求解方法, 采用大量数值算例反复求解, 并分别与度约束最小树和多目标最小树等相关问题计算结果进行比较, 有效避免了生成树某些顶点度数过大、过于集中而造成的不稳定问题, 并能得到多个 Pareto 有效解, 效果令人满意。尤其是当问题规模较大时, 本文算法仍能在短时间内给出理想的结果, 这在一些需要快速、实时响应的大型系统中具有明显优势。

4 结束语

多目标 MIN-MAX 度最小树问题由多目标最小树问题和 MIN-MAX 度生成树问题以及度约束最小树等问题延伸扩展而来, 在现实生活中有着一系列的应用。本文设计了相应的快速优化算法进行求解, 取得了满意的效果。特别是本文算法思想还可面向一些对最大顶点度极度敏感的问题, 或动态变化环境下需快速响应的实时动态网络规划问题等, 后续将研究更完善的求解方案和技术, 探索对特定实际网络图的解决方案。

参考文献:

[1] JOTHI R, MOHANTY S K, OJHA A. Fast approximate

minimum spanning tree based clustering algorithm[J]. Neurocomputing, 2018, 272: 542–557.

[2] GAO X, JIA L F, KAR S. Degree-constrained minimum spanning tree problem of uncertain random network[J]. Journal of Ambient Intelligence and Humanized Computing, 2017, 8(5): 747–757.

[3] CLÍMACO J C N, CAPTIVO M E, PASCOAL M M B. On the bicriterion-minimal cost/minimal label - spanning tree problem[J]. European Journal of Operational Research, 2010, 204(2): 199–205.

[4] ZHOU G, GEN M. Approach to the degree-constrained minimum spanning tree problem using genetic algorithm[J]. Engineering Design and Automation, 1997, 3(2): 157–165.

[5] 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.

[6] JOTHI R, RAGHAVACHARI B. Degree-bounded minimum spanning trees[J]. Discrete Applied Mathematics, 2009, 157(5): 960–970.

[7] DE ALMEIDA A M, MARTINS P, DE SOUZA M C. Min-degree constrained minimum spanning tree problem: complexity, properties, and formulations[J]. International Transactions in Operational Research, 2012, 19(3): 323–352.

[8] 马良, 蒋馥. 度限制最小树的蚂蚁算法[J]. 系统工程学报, 1999, 14(3): 211–214.

[9] 孙治国, 滕弘飞. 布线设计的模型和算法研究进展[J]. 系统工程与电子技术, 2003, 25(12): 1536–1541.

[10] PREMKUMAR G, CHU C H, CHOU H H. Telecommunications network design-comparison of alternative approaches[J]. Decision Sciences, 2000, 31(2): 483–506.

[11] 史小红, 贾新娟, 王燕. 基于 Hamilton 路模型的蛋白质结构预测的研究[J]. 数学的实践与认识, 2009, 39(22): 100–104.

[12] 温雪莲, 娄定俊, 陆芸婷, 等. 求 Halin 图中给定两点之间最优 Hamilton 路的有效算法[J]. 计算机科学, 2007, 34(9): 176–180, 217.

[13] ANDERSEN K A, JÖRNSTEN K, LIND M. On bicriterion minimal spanning trees: an approximation[J]. Computers & Operations Research, 1996, 23(12): 1171–1182.

(编辑: 丁红艺)