文章编号：2095-2163(2019)06-0001-06 中图分类号：TP391.41 文献标志码：A

# 不确定图最小生成树算法

张安珍，李建中
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

**摘 要：**很多领域产生的大量数据都可以很自然地用不确定图模型表示和描述，如蛋白质交互网络、社交网络、无线传感器网络等。本文研究不确定图上最可靠的最小生成树问题，该问题具有广泛的应用价值和研究意义。精确地求解算法需要枚举所有可能的最小生成树并找出其中出现次数最多的那个。因此，枚举开销随着边数增多呈指数增长，当图规模较大时并不可行。为此本文提出了一个时间复杂度为 $O(d\,|V|^2)$ 的启发式贪心算法，其中 $d$ 为最大的顶点度数，$|V|$ 为顶点数。实验结果表明，该算法具有较好的效率和较高扩展性。

**关键词：**不确定图；最可靠最小生成树；贪心算法

## Minimum spanning tree on uncertain graphs

ZHANG Anzhen, LI Jianzhong
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China)

**【Abstract】** In recent years, lots of data in various domain can be represented and described by uncertain graph model, e.g. protein interaction networks, social networks, wireless sensor networks, etc. This paper investigates the most reliable minimum spanning tree on uncertain graph model, which has wide applications and research significance. The accurate algorithm enumerates all possible minimum spanning trees and find the most frequent one, the enumeration cost increases exponentially as the edge grows, which cannot fit the large graph computation. A heuristic greedy algorithm in $O(d\,|v|^2)$ was proposed to solve the problem, where $d$ is the largest degree, and $|V|$ is the number of vertexes. Experimental results show that the algorithm is both efficient and scalable.

**【Key words】** uncertain graph; most reliable minimum spanning tree; greedy algorithm

## 0 引 言

近年来，很多领域产生的大量数据可以利用图模型表示。由于测量数据的工具不精确、数据本身性质等多方面原因导致图数据普遍存在不确定性[1]，如蛋白质交互网络（PPI）、社交网络、无线传感器网络等，将不确定性融入到图中得到不确定图。

确定图上的最小生成树问题具有十分重要的实际意义。在连通图 $G=(V,E)$ 中，$E$ 中每条边有对应的权值，图 $G$ 的一棵生成树是连接 $V$ 中所有顶点的一棵树，生成树中所有边的权值总和称为生成树的代价，使这个代价最小的生成树称为图 $G$ 的最小生成树（Minimum Spanning Tree，MST）[2-3]。很多实际应用问题可以通过求解最小生成树得到很好的解决。例如，道路规划问题，可将各个城镇作为图的顶点，城镇之间的道路作为边，每条道路的修建代价作为边上的权值，在该图上求最小生成树可以得到修建代价最少的修路方案。如，在无线传感器网络中，由于传感器节点间的通信链路存在一定的干扰，每条链路存在一定的失效风险，因此其拓扑结构可以很自然地建模为不确定图。顶点表示各传感器节点，边表示可能存在的通信链路，边的权值由距离决定，边的存在概率表示该链路正常工作的可能性大小。在该图上求解最可靠的最小生成树非常重要，其上的边集即链路集合是最稳定可靠的连通路径，利用该生成树可以优化路由选路过程，用最少的代价完成全部节点间通信的同时，保证了网络通信的可靠性与稳定性。

不确定图是边带有概率的特殊加权图，边的概率表示该边存在的可能性大小。因此不确定图有多种可能的存在形式，每一个存在形式称为蕴含子图[4]，蕴含子图是一个以一定概率确定存在的加权图，由此可知每个连通的蕴含子图上都有至少一棵最小生成树，这些最小生成树构成了不确定图的最小生成树集合。

本文研究不确定图上最可靠的最小生成树问

**基金项目：**国家自然科学基金（61190115，61033015）；国家重点基础研究发展计划（973）（2012CB316200）。

**作者简介：**张安珍（1990-），女，博士研究生，主要研究方向：数据质量、海量数据计算；李建中（1950-），男，教授，博士生导师，主要研究方向：并行数据库、传感器网络、海量数据管理等。

**收稿日期：**2018-11-15

题,即求解最小生成树集合中出现次数最多的最小生成树,记为 $MST_{max}$。精确的求解 $MST_{max}$ 需要枚举所有连通的蕴含子图,并在其上计算最小生成树,在得到的最小生成树集合中选取出现次数最多的即为 $MST_{max}$。然而枚举所有蕴含子图需要 $O(2^{|E|})$ 的时间开销,其中 $|E|$ 表示不确定图上的边集大小,在实际应用中图规模往往较大,精确算法并不可行。为此,本文提出了一个时间复杂度为 $O(d\ |V|^2)$ 的启发式贪心算法近似求解 $MST_{max}$,其中 $d$ 表示顶点的最大度数,$|V|$ 表示顶点大小,通过实验结果分析,发现该算法能够在大多数情况下找到 $MST_{max}$。

## 1 问题定义

本节介绍一种不确定图模型,并形式化描述不确定图上的最可靠的最小生成树问题。

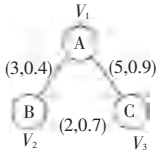
### 1.1 不确定图模型

不确定图是一个四元组 $G=(V,E,P,W)$,其中 $V$ 是顶点集,$E$ 为边集,$P$ 是边的存在概率函数,即 $E$ 到(0,1]的映射,$P(e)$ 表示边 $e$ 存在的概率,$W$ 为权重函数,定义了每条边的权值大小。由于边的不确定性导致不确定图具有多种存在形式,每种确定的存在形式称之为蕴含子图,也称为可能世界。所有的蕴含子图构成不确定图所蕴含的确定图集合,记为 $Imp(G)$。对于 $g \in Imp(G)$,称之为 $G$ 蕴含 $g$,记作 $G \Rightarrow g$,假定所有边相互独立,蕴含概率 $P(G \Rightarrow g)$ 等于所有 $g$ 中的边的存在概率与不在 $g$ 中的边不存在概率之积,如公式(1)所示。

$$P(G \Rightarrow g) = \prod_{e \in E_g} p(e) \cdot \prod_{e \in E \setminus E_g} (1 - p(e)). \quad (1)$$

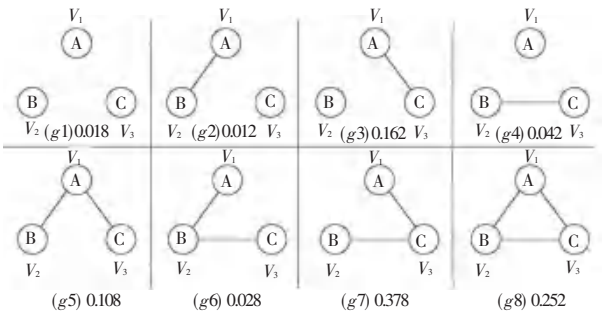
下面通过一个简单的例子来说明蕴含子图及其存在概率的含义。

**例 1** 如图 1(a)所示,图 $G$ 是包含 3 个顶点和 3 条边的不确定图,每条边存在与否决定了图 $G$ 共有 $2^3 = 8$ 个蕴含子图,每个子图的存在概率由公式(1)计算得到。以图 1(b)中的子图 g2 为例,$P(G \Rightarrow g2) = 0.4 \times (1 - 0.9) \times (1 - 0.7) = 0.012$,其它子图的存在概率计算与之类似,结果如图 1($b$)所示。




(a) 一个简单的不确定图 $G$

(a) A simple uncertain graph $G$




(b) $G$ 的全部蕴含子图

(b) All $Imp(G)$

**图 1 不确定图 $G$ 及其蕴含子图**

**Fig. 1 Uncertain graph G and $Imp(G)$**

### 1.2 最可靠的最小生成树

最可靠的最小生成树 $MST_{max}$ 是最小生成树集合中出现次数最多的,也就是存在概率最大的最小生成树,代表了不确定图上最稳定可靠的连通结构,具有广泛的应用价值。最小生成树 $MST_i$ 的存在概率 $P(MST_i)$ 由其所在的蕴含子图的存在概率决定,等于所有以 $MST_i$ 为其上最小生成树的蕴含子图的存在概率之和,$P(MST_{max})$ 是最小生成树集合中存在概率最大的,$MST_{max}$ 的形式化定义如图 2 所示。

$$MST_{max} = argmax\{P(MST_i) = \sum_{g \in Imp(G)} P(G \Rightarrow g)\ I_1(g)\}, \quad (2)$$

其中,$I_1$ 是指示函数,表明 $MST_i$ 是否是 $g$ 上的最小生成树。

$$I_1 = \begin{cases} 1, & MST_i \text{ 是 } g \text{ 的最小生成树} \\ 0, & MST_i \text{ 不是 } g \text{ 的最小生成树}, \end{cases} \quad (3)$$

下面通过例 2 求图 1(a)中不确定图 G 上的 $MST_{max}$ 来说明公式(3)的含义。

**例 2** 图 $G$ 有 8 个蕴含子图如图 1(b)所示,只有 4 个是连通的,分别是 $g5$、$g6$、$g7$ 和 $g8$,分别求其上的最小生成树,其中 $g8$ 上的最小生成树和 $g6$ 相同。因此共有 3 个不同的最小生成树,如图 2 所示,分别是 $MST_1$、$MST_2$ 和 $MST_3$。这 3 棵生成树的存在概率依次为 $P(MST_1) = P(G \Rightarrow g5) = 0.108$,$P(MST_2) = P(G \Rightarrow g6) + P(G \Rightarrow g8) = 0.028 + 0.252 = 0.28$,$P(MST_3) = P(G \Rightarrow g7) = 0.378$。

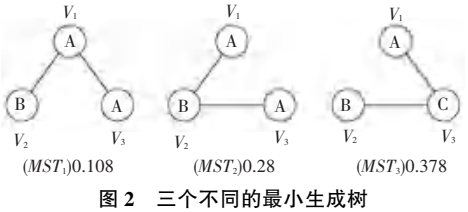



**图 2 三个不同的最小生成树**

**Fig. 2 Three different *MST***

$MST_{max}$ 是最小生成树集合中存在概率最大的那棵，即 $MST_3$，$MST_{max}=\{AC,BC\}$，$P(MST_{max})=P(MST_3)=0.378$，利用 $|MST|$ 表示最小生成树的代价，则 $|MST_{max}|=|MST_3|=7$。

由 $MST_{max}$ 的定义可知，一种直观的求解算法是枚举所有蕴含子图 $g$，并在 $g$ 上求最小生成树得到一个最小生成树集合，求集合中存在概率最大的最小生成树。然而该算法在实践上并不可行，图 G 共有 $2^{|E|}$ 个蕴含子图，枚举代价随边数增多呈指数增长，当图规模较大时，算法效率十分不理想。因此本文提出一种启发式的贪心算法近似求解 $MST_{max}$，时间复杂度为 $O(d^2|V|^2)$，相对于枚举法性能大大提高，实验结果表明该算法在通常情况下能够得到 $MST_{max}$。

## 2　贪心算法求最可靠最小生成树

启发式贪心策略近似求解不确定图上最可靠的最小生成树，在 2.1 节给出算法的详细描述，而后在 2.2 节分析贪心算法的运行时间。

### 2.1　算法描述

给定连通不确定图 $G=(V,E,P,W)$，求解其上 $MST_{max}$ 的算法思想与 Prim 算法求解确定图上最小生成树类似，维持一棵树 $A$，树 $A$ 从任意根顶点 $r$ 开始形成，并逐渐生成，直至树 $A$ 覆盖了 $V$ 中的所有顶点，每一次，一条连接树 $A$ 与 $G_A=(V,A)$ 中某孤立顶点的概率最大的轻边被加入到树 $A$ 中，其中轻边是指权值最小的边，下面给出概率最大轻边的定义。

**定义 1**　概率最大的轻边（Light Edge with Maximum Probability，LEMP）. 将所有连接树 $A$ 与森林 $G_A=(V,A)$ 中某孤立顶点的边加入队列 $S$ 中，按公式(4) 计算其加入到树 $A$ 中的概率，其中概率值最大的边即为 $LEMP$。

将队列 $S$ 中的边按权值非降序排列，$p_i$ 表示位于排序队列中第 $i$ 条边的存在概率，$1\leqslant i\leqslant|S|$，则第 $i$ 条边作为概率最大的轻边（$LEMP$）加入到树 $A$ 中的概率 $P_i$ 为：

$$P_i=\Big(\prod_{j=1}^{i-n_i-1}(1-p_j)\Big)\cdot p_i. \tag{4}$$

其中，$n_i$ 表示队列中所有与第 $i$ 条边权值相等，但位置排在其前面的边的个数，$0\leqslant n_i\leqslant i-1$。下面举例说明公式(4)。

**例 3**　假设 $G$ 中所有与 $A$ 中顶点直接相连但不在 $A$ 中的边有 $\{e_1,e_2,e_3\}$，将其加入队列 $S$，按照权值升序排列，假设 $w_1<w_2=w_3$，则这 3 条边加入到树 $A$ 中的概率依次为：$p_1$，$(1-p_1)\cdot p_2$，$(1-p_1)\cdot p_3$。

直观地讲，$e_1$ 具有最小权重值 $w_1$，故其作为轻边加入树 A 的概率就是其自身的存在概率 $p_1$；对于 $e_2$，由于其的权值 $w_2$ 大于 $e_1$ 的权值 $w_1$，所以只有在 $e_1$ 不存在的情况下，$e_2$ 才能作为轻边加入到 $A$ 中，概率为 $e_1$ 不存在的概率乘以 $e_2$ 存在的概率，即 $(1-p_1)\cdot p_2$；对于 $e_3$，由于 $e_3$ 的权值 $w_3$ 和 $e_2$ 的权值 $w_2$ 相等，则其加入到 $A$ 的概率与 $e_2$ 存在与否无关，概率等于 $e_1$ 不存在的概率乘以 $e_3$ 存在的概率，即 $(1-p_1)\cdot p_3$。

下面给出具体的算法描述，见算法 1。实现时，采用插入排序法排序队列 $S$，因为每次添加新边之前，$S$ 中已有的边是按权值升序排列的，只需要将新加入的边插入到已经排好序的队列中即可，这样减少了全部边重新排序的时间开销。

**算法 1**　最可靠的最小生成树算法.

输入：不确定图 $G=(V,E,P,W)$。

输出：最可靠的最小生成树 $MST\_max$ 的边集 $A$

(1) $MST\_V=\{r\}$；$A=\phi$；$S=\phi$

(2) 将所有与 $r$ 相连的边添加到队列 $S$ 中

(3) WHILE $|MST\_V|<|V|$ DO

(4) 将 $S$ 中的边按照权值升序排列

(5) 按照公式(4)计算 $S$ 中每条边加入 $A$ 的概率值

(6) 利用最大堆 $H$ 求概率最大的边，记为 $(u,v)$

(7) 将 $(u,v)$ 加入 $A$ 中，将不在 $MST\_V$ 中的顶点

(8) 假设是 $v$，加入到 $MST\_V$ 中

(9) 删除 $S$ 中所有与 $v$ 相连的边

(10) 将 $G$ 中所有与 $v$ 相连但另一端的顶点不在 $MST\_V$ 中的边加入到 $S$ 中

(11) END WHILE

### 2.2　算法的分析

下面分析最坏情况下算法的运行时间。将图 $G$ 中最大顶点度数记作 $d$，$1\leqslant d\leqslant|V-1|$，第 $i$ 次迭代时队列 $S_i$ 的长度记为 $|S_i|$，则有如下关系成立：

$$\begin{cases}|S_1|\leqslant d, i=0,\\ |S_{i+1}|\leqslant(|S_i|-1)+(d-1), i\geqslant 1.\end{cases} \tag{5}$$

第一次迭代时，$A$ 中只有一个顶点 $r$，将 $r$ 的邻边加入到 $S_1$ 中，最多加入 $d$ 条边；第 $i+1$ 次迭代时，$S_{i+1}$ 中的边由 $S_i$ 去掉包含新顶点 $v$ 的边后，再并上 $G$ 中 $v$ 的邻边中另外一端不在 $A$ 中的边，其中 $S_i$ 最少去掉

一条边$\{u,v\}$,$G$ 中 $v$ 的邻边集合最多加入 $d-1$ 条边,即除了$\{u,v\}$ 以外的全部邻边,故$|S_{i+1}|$满足式(5)。由公式(5) 经过等差数列求通项得到$|S_i|$£$(d-2)i+2=O(di)$。

接下来分析 while 循环,迭代总共执行了 $|V|$ 次,在第 $i$ 次迭代时 $|S_i|=O(di)$;第 3 步排序 $S_i$ 采用插入排序,最坏情况即新加入队列 $S_i$ 中的边的权值都小于 $S_{i-1}$ 中的边,此时需要比较的次数为 $O(|S_{i-1}|\times d)=O(d^2i)$;第 4 步利用最大堆求 $P$ 值最大的边需要 $lg|S_i|=O(lgdi)$;第 5 步将新边 $\{u,v\}$ 加入 $A$ 中需要 $O(1)$ 时间;第 6 步删除操作需要扫描一遍队列,故需要 $O(|S_i|)=O(di)$ 时间;第 7 步加入新边,最多添加 $d-1$ 条,故需要 $O(d)$ 时间,总的运行时间 $T(n)=\sum_{i=1}^{|V|}(O(d^2i)+O(lgdi)+O(1)+O(di)+O(d))=O(d^2|V|^2)$,可见算法的运行时间主要由顶点数和最大顶点度数决定。

## 3 实验验证

### 3.1 实验环境及实验数据

为了验证启发式贪心算法的效率以及有效性,分别在合成数据集和人造数据集上进行测试。采用 C++编程,实验环境为 PC 机,英特尔(R)酷睿 2 双核 2.4 GHz CPU 和 2 GB 的主内存,运行系统为 Ubutu 12.04。MapReduce 算法在完全分布式环境下测试,Hadoop 版本为 2.6.0,实验环境为 3 台 PC 机,配置为英特尔(R)酷睿 TM 四核 3.4 GHz CPU 和 32GB 内存。

实验中的合成数据集是在真实不确定图数据上,随机标注权值得到,权值取 0~100 之间的整数。使用文献[6]提供的 2 个真实的不确定图数据集,Nature 是蛋白质交互网络,Flickr 是一个社交网络,图规模及连通性见表 1,其中 N(MST)表示 $MSF_{max}$ 中包含的连通分支数目。

**表 1 合成数据集**

**Tab. 1 Synthesis data set**

| 不确定图 | 图规模$(V,E)$ | 连通 | $N(MST)$ |
|---|---|---|---|
| Nature | (2 708,7 123) | 否 | 63 |
| Flickr | (21 594,108 258) | 否 | 1 732 |

人造数据集采用随机生成一些不确定图,顶点之间的边及边上的权值和概率都随机生成,权值取 0~100 之间的整数,概率取值 0.00~0.99 之间的两位有效小数。在人造数据集 1 中,控制顶点的平均度数 $d'$ 相同,都为 1.23,顶点大小从 1k 递增到 10k,每次递增 1k;在人造数据集 2 中,控制顶点大小一定,都为 1k,平均度数取 3~7.5,每次递增 0.5。人造数据集 3 是一个小规模图集合,顶点大小从 10 递增至 50,平均度数都为 3。

### 3.2 实验结果

贪心算法在求 $MST_{max}$ 时,如果图不连通,则求最可靠的最小生成森林 $MSF_{max}$。具体来说,当算法求得一棵生成树 $A$ 时,若 $A$ 中顶点数小于图 $G$ 的顶点数$|V|$,则将$A$加入到$MSF_{max}$中,同时随机选取一个不在 $A$ 中的顶点作为新的根节点,生成另外一棵生成树,直至$MSF_{max}$ 覆盖所有 $G$ 中的全部顶点。

首先测试贪心算法在不同图规模下的性能表现。在合成数据集 Nature 和 Flickr 上,逐步抽取其上的子图进行测试,子图大小为原图的 10%~100%,运行时间随顶点大小的变化如图 3 所示。实验结果表明运行时间随顶点数 $|V|$ 的增加大致呈抛物线增长趋势,与上一节分析的算法运行时间 $O(d^2|V|^2)$ 一致,另外在 Flickr 数据集上,图规模为(17 273, 102 356)时的运行时间比图规模为(15 113,97 743)反而少,这是因为(17 273,102 356)的平均度数 $d'$ 为 5.9,比(15 113,97 743)上的平均度数 6.4 小,接下来测试平均顶点度数对运行时间的影响。

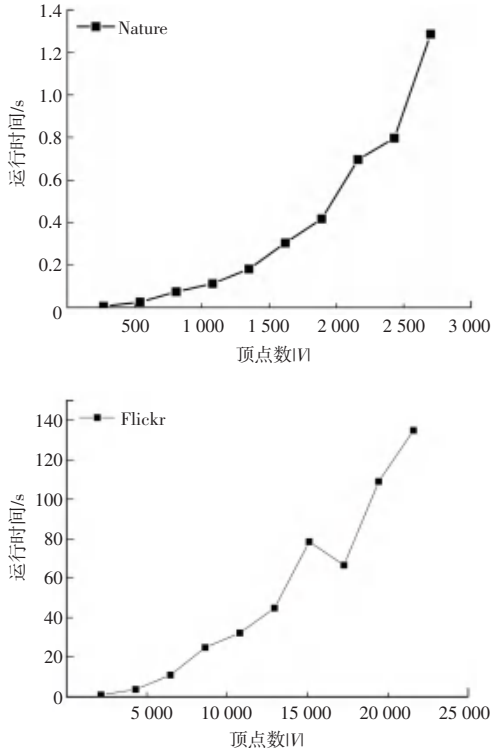
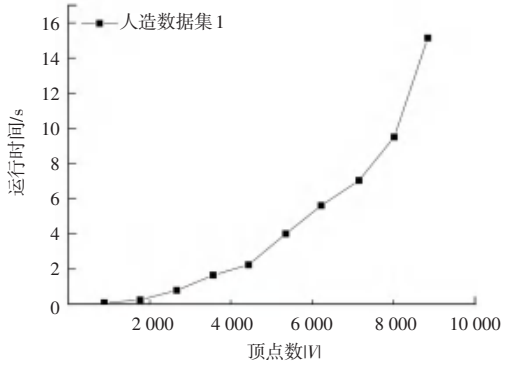



**图 3 不同顶点数$|V|$下的运行时间**

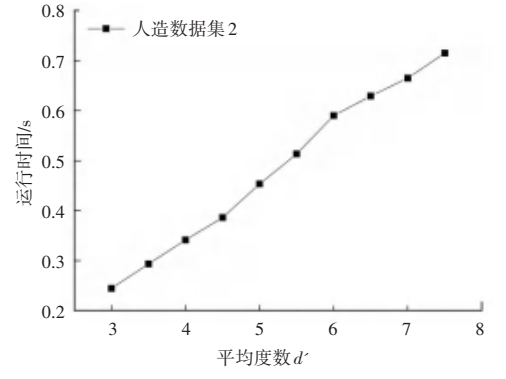
**Fig. 3 Execution Time with Different $|V|$**

在人造数据集 1 中，顶点平均度数 $d'$ 一定，都为 1.23，在其上运行贪心算法，测试顶点大小对运行时间的影响，结果如图 4(a)所示。在人造数据集 2 上，顶点大小固定，都为 1k，边的大小从 3k 递增到 7.5k，即顶点平均度数取 3~7.5，运行结果如图 4(b)所示。




(a) $d'$ 不变

(a) Fix $d'$




(b) 顶点数 $|V|$ 不变

(b) Fix $|V|$

**图 4　平均度数 $d'$ 对运行时间影响**

**Fig. 4　Execution time with different $d'$**

由图 4(a)分析可知，当平均顶点度数 $d'$ 一定时，算法的运行时间随 $|V|$ 的增大呈抛物线增长趋势，并且期间没有异常点出现。图 4(b)中的实验结果表明，当顶点数一定时，运行时间随平均度数 $d'$ 增加而呈线性增长。

若要验证贪心算法得到的最小生成树与精确解得到的最小生成树是否一致，需要枚举所有可能的最小生成树，枚举代价非常大，为 $O(2^{|E|})$，这为验证工作增加了很大困难，因此需要设计一个随机算法用来对比实验结果。

随机算法每次向树 $A$ 中随机添加队列 $S$ 中的一条边，由于最小生成树的存在概率等于每次迭代时加入 $A$ 中边的概率 $P$ 累乘，当图规模太大时，最小生成树的概率 $P$ 很小，为了方便实验数据对比，取其对数得到 $\log P$ 从而将其放大，由于 log 函数具有单调递增性，所以不会影响实验对比结果。在人造数据集 3 小图上进行实验，随机算法执行 3 次，实验结果如图 5 所示。

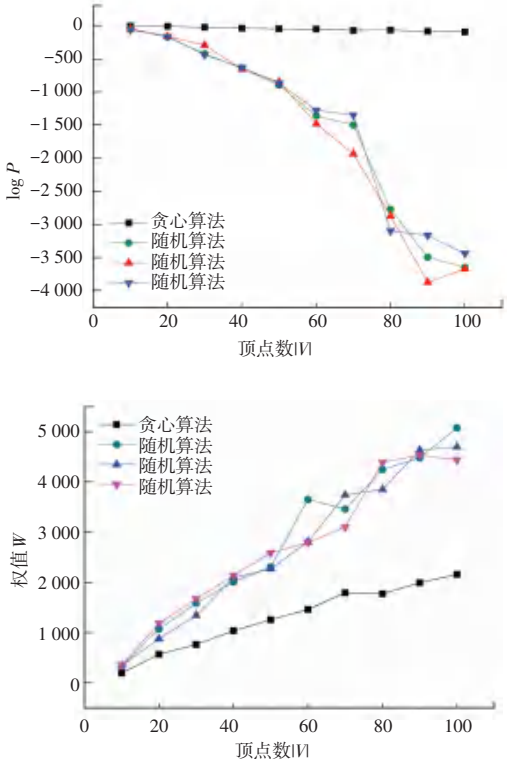



**图 5　2 种算法运行结果比较**

**Fig. 5　Comparison between two algorithms**

由图 5 上的结果可见，随着顶点数增多，贪心算法求出的最小生成树的概率比 3 次随机算法求得的最小生成树概率大很多，并且权值更小，这在很大程度上说明了贪心算法求出的 $MST_{max}$ 是精确解的一个很好的近似。

## 4　结束语

本文研究了不确定图上最可靠的最小生成树问题。最可靠的最小生成树代表了不确定图上最稳定的连通分支，具有广泛的应用价值。精确求解算法的时间开销非常大，因此本文给出一个多项式时间的启发式贪心算法，实验结果表明该算法在大多数情况下能够得到最优解。

## 参考文献

[1] Potamias M, Bonchi F, Gionis A, et al. K-nearest neighbors in uncertain graphs [J]. Proceedings of the VLDB Endowment, 2010, 3(1-2): 997-1008.

[2] Prim R C. Shortest connection networks and some generalizations [J]. Bell system technical journal, 1957, 36(6): 1389-1401.

(下转第 12 页)