学校代码 10530　　学　　号 201230111747

分 类 号 TP311　　密　　级

湘潭大学

硕 士 学 位 论 文

最小生成树在不确定图中的应用与研究

学 位 申 请 人 唐 杰

指 导 老 师 文 中 华 教授

学 院 名 称 信息工程学院

学 科 专 业 计算机科学与技术

研 究 方 向 不 确 定 图

二〇一四年十一月三十日

最小生成树在不确定图中的应用与研究

学 位 申 请 人 唐 杰

导师姓名及职称 文 中 华 教授

学 院 名 称 信息工程学院

学 科 专 业 算机科学与技术

研 究 方 向 不 确 定 图

学位申请级别 工 学 硕 士

学位授予单位 湘 潭 大 学

论文提交日期 2014-11-30

Application and Research of Minimum Spanning Tree in Uncertain Graph

Candidate Jie Tang

Supervisor Professor Zhonghua Wen

College College of Information Engineering

Program Computer Science and Technology

Specialization Uncertain Graph

Degree Master of Science

University Xiangtan University

Date April 20th, 2012

湘潭大学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：　　日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湘潭大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

涉密论文按学校规定处理。

作者签名：　　日期：　　年　　月　　日

导师签名：　　日期：　　年　　月　　日

目 录

第一章　绪 论

第二章　不确定图中最优生成树和次优生成树算法

§2.1　问题定义

定义 2.1.1 (不确定图) 不确定图是一个四元组$\mathcal{G} = (V, E, W, P)$，其中$V$是顶点集，$E$是边集，$W = \{w(e) \,|\, e \in E, w(e) \in \mathbb{N}^+\}$是边的权重集，$P = \{p(e) \,|\, e \in E, p(e) \in (0, 1]\}$是边存在可能性的集合。

定义 2.1.2 (蕴含图) 令不确定图$\mathcal{G} = (V, E, W, P)$，若确定图$G = (V_G, E_G, W_G)$是$\mathcal{G}$的一个蕴含图，则必然满足$V_G = V$，$E_G \subseteq E$和$W_G = \{w(e)|e \in E_G\} \subseteq W$。

从定义 2.1.1可知，不确定图$\mathcal{G} = (V, E, W, P)$每条边以$p(e)$的概率存在，在可能世界模型下，每条边有存在和不存在两种可能性，所以可以派生出$2^{|E|}$个蕴含图。本文沿用文献[5, 8, 12]对不确定图模型所做的假设，即不确定图中不同边的概率分布相互独立。将蕴含图G和不确定图$\mathcal{G}$之间的关系表示为$\mathcal{G} \Rightarrow G$。基于以上假设，蕴含图$G$存在的概率为

$$\Pr(\mathcal{G} \Rightarrow G) = \prod_{e \in E_G} p(e) \prod_{e \in E \setminus E_G} (1 - p(e)). \tag{2.1}$$

记不确定图$\mathcal{G}$的所有蕴含图的集合为$Imp(\mathcal{G})$。由文献[6]可知$\mathcal{G}$中所有蕴含图出现的概率和为1，即

$$\sum_{G \in Imp(\mathcal{G})} Pr(\mathcal{G} \Rightarrow G) = 1. \tag{2.2}$$

例 1 图 *2.1(a)*为一个不确定图$\mathcal{G}_1$，边上的两个数字分别代表权值和概率，由于该不确定图有五条可能出现的边，因此该不确定图有2^5个蕴含图。图*2.1(b)*为不确定图$\mathcal{G}_1$的一个蕴含图，显然该蕴含图存在的概率为$p(e_2) \times p(e_3) \times p(e_4) \times p(e_5) \times (1 - p(e_1))) = 0.03528$。

在传统图论中，图的最小生成树被定义为边的权值和最小的生成树，一个图可能存在多个最小生成树，这些最小生成树具有相同的权值和。然而在不确定图中，每条边都有一个存在的概率，这样会导致每一颗最小生成树都只有一定的概率存在，我们可以通过一个稳定性来区分最小生成树的好坏，即

$$R_T = \prod_{e \in E_T} p(e) \tag{2.3}$$

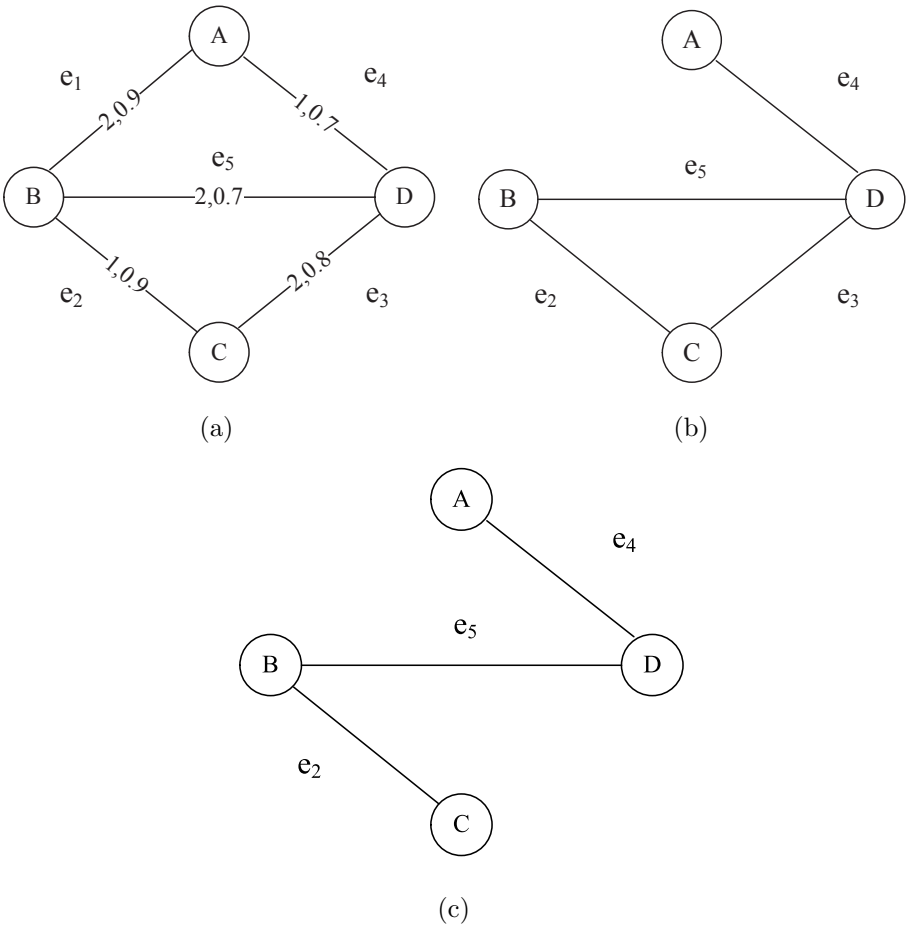

图 2.1 Example 1: (a) uncertain graph $\mathcal{G}_1$; (b) an implicated graph from $\mathcal{G}_1$; (c) main implicated graph from $\mathcal{G}_1$.

公式 2.3的含义与公式 2.1有所不同，公式 2.3的另一层含义可以解释为所有包含最小生成树T的蕴含图存在的概率和。

为了区分每颗生成树，我们需要给每个生成树进行编号，假设所有生成树的边都按照编号从小到大进行排序，我们称字典序较小的生成树具有更小的编号。

定义 2.1.3 (最小生成树) 设不确定图$\mathcal{G} = (V, E, W, P)$，若生成树$T$满足$\forall T'(\sum_{e\in E_T} W(e) \leq \sum_{e'\in E_{T'}} W(e'))$，则称$T$为不确定图$\mathcal{G}$的最小生成树，记编号最小的最小生成树为$T_M^{\mathcal{G}}$，其边集为$E_M^{\mathcal{G}}$，边的权值和为$W_M^{\mathcal{G}}$。

定义 2.1.4 (最小最大乘积生成树) 设不确定图$\mathcal{G} = (V, E, W, P)$，若生成树$T$满足$\forall T'(\prod_{e\in E_T} P(e) \leq \prod_{e'\in E_{T'}} P(e'))$，则称$T$为不确定图$\mathcal{G}$的最小乘积生成树，记编

号最小的最小乘积生成树为$T_{PL}^{\mathcal{G}}$，其边集为$E_{PL}^{\mathcal{G}}$，边的权值和为$W_{PL}^{\mathcal{G}}$。若生成树T满足$\forall T'(\prod_{e\in E_T} P(e) \geq \prod_{e'\in E_{T'}} P(e'))$，则称$T$为不确定图$\mathcal{G}$的最大乘积生成树，记编号最小的最大乘积生成树为$T_{PB}^{\mathcal{G}}$，其边集为$E_{PB}^{\mathcal{G}}$，边的权值和为$W_{PB}^{\mathcal{G}}$

定义 2.1.5 (最优生成树) 设$Imp(\mathcal{T})$为不确定图$\mathcal{G}$中所有最小生成树的集合，若最小生成树T满足$\forall T' \in Imp(\mathcal{T})(R_{T'} \leq R_T)$，则称$T$为不确定图$\mathcal{G}$的最优生成树。记编号最小的最优生成树为$T_O^{\mathcal{G}}$，其边集为$E_O^{\mathcal{G}}$，边的权值和为$W_O^{\mathcal{G}}$。

显然，最优生成树不是唯一的，我们也可以将最优生成树看作不确定图中的最小生成树中的最小乘积生成树。

§2.2 最优生成树算法

定理 2.2.1 设不确定图$\mathcal{G} = (V, E, W, P)$和$\mathcal{G}' = (V, E, W, P')$，其满足$P'(e) = \log_2(P(e))$，则$W_{PL}^{\mathcal{G}} = 2^{W_M^{\mathcal{G}'}}$。

证 明 设T为$\mathcal{G}$和$\mathcal{G}'$的任意一颗生成树，其边集为E_T，则$\sum_{e\in E_T} P'(e) = \log_2 \prod_{e\in E_T} P(e)$，显然生成树$T$在$\mathcal{G}'$中的边权和与在$\mathcal{G}$中边权的乘积成正比，当$T$为$\mathcal{G}'$的最小生成树时，则有$W_M^{\mathcal{G}'} = \log_2 W_{PL}^{\mathcal{G}}$，即$W_{PL}^{\mathcal{G}} = 2^{W_M^{\mathcal{G}'}}$。 ∎

由定理 2.2.1可知，要想求得最小乘积生成树，可以将图中所有边的权值进行log变换，然后求新图的最小生成树。我们将在后面使用该思路去求解不确定图的最优生成树。

推论 2.2.1 若生成树的边权和是最小的，那么其边权的乘积也是最小的；若生成树的边权和是最大的，那么其边权的乘积也是最大的。

我们知道使用$kruskal$算法求解最小生成树的第一步是需要对图中所有的边进行排序，因此可以假设不确定图$\mathcal{G} = (V, E, W, P)$中所有的边按照如下优先级进行排序：

- 将权值较小的边排在前面；
- 对于权值相同的边，则将概率较大的边排在前面；

- 若两者的值都相同，则将编号较小的边排在前面。

我们称靠前的边具有更高的优先级，最优生成树的算法步聚如下：

1. 新建一个不确定图$\mathcal{G}'$，其顶点集和不确定图$\mathcal{G}$相同，边集为空；
2. 假设不确定图$\mathcal{G}$的边集已经按照上面的规则进行排序；
3. 依次处理排序好的边，若这条边连接的两个顶点不在同一个连通分量中，则将这条边加入不确定图$\mathcal{G}'$;
4. 重复第3步，直到不确定图$\mathcal{G}'$中所有的顶点都连通。

这样，我们得到的不确定图$\mathcal{G}'$既是不确定图$\mathcal{G}$的最优生成树, 显然该最优生成树也是编号最小的最优生成树$T_O^{\mathcal{G}}$。

定理 2.2.2 *最优生成树算法是正确的。*

证 明 显然最优生成树的算法步聚和$Kruskal$算法的步聚一致，因此我们得到的生成树一定是最小生成树。根据最小生成树的性质，所有的最小生成树应该具有相同的边权值序列（假设边权值按照从小到大排序），并且相同权值的边构成的连通分量应该相同，根据最优生成树算法可知，由相同权值的边构成的连通分量，其边的概率和是最大的。再由推论 2.2.1可知，这些边的概率乘积也是最大的，因此最优生成树算法是正确的。 ∎

例 2 *图 §2.2为一个不确定图$\mathcal{G}$，按照优先级排序后的边集为$\{e_1, e_2, \cdots, e_{10}\}$。图 §2.2为图 §2.2中不确定图的最优生成树求解过程，红色标记的边为加入生成树中的边，在图(2)到图(3)时，由于顶点D和顶点E已经在同一个连通分量，所以边e_3没有加入生成树的边集中，图(6)中红线构成的生成树则为最优生成树。从图中可知，$W_O^{\mathcal{G}} = 13$且$R_{T_O^{\mathcal{G}}} = 0.14112$。*

算法 1为最优生成树算法的伪代码，我们使用了并查集数据结构来快速合并两个连通分量，每个连通分量都会有一个唯一的根节点，通过函数GetRoot()即能快速查询到每个顶点的根节点。合并连通分量只需要一条赋值语句即可，函数OST()中的第7行则是合并顶点u和顶点v所在的连通分量。

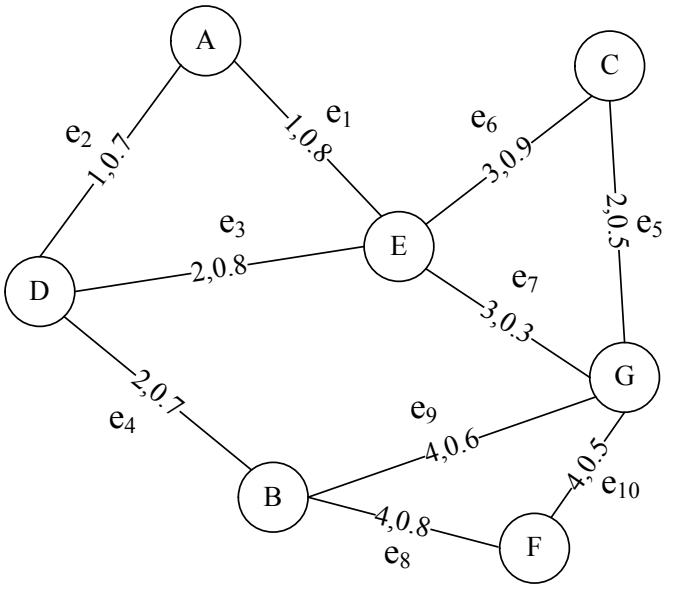

图 2.2 Example 2: 不确定图$\mathcal{G}$。

Algorithm 1 最优生成树算法

模块1： 并查集

initialization $RT_i \leftarrow i(i = 1, 2, \cdots, |V|)$

procedure GetRoot(u)
  if $u = RT_u$ **then return** u
  else return $(RT_u \leftarrow \text{GetRoot}(RT_u))$

模块2： 最优生成树算法

procedure OST($\mathcal{G}$) ▷ $\mathcal{G} = (V, E, W, P)$
  $E_O^{\mathcal{G}} = \emptyset$
  for each $e_i \leftarrow (u, v) \in E$ **do**
    $ru \leftarrow \text{GetRoot}(u)$
    $rv \leftarrow \text{GetRoot}(v)$
    if $ru \neq rv$ **then**
      $RT_{ru} = rv$
      $E_O^{\mathcal{G}} = E_O^{\mathcal{G}} \cup \{e_i\}$
  return $E_O^{\mathcal{G}}$

§2.3 次优生成树算法

定义 2.3.1 (次优生成树) 设$Imp(\mathcal{T})$为不确定图$\mathcal{G}$中所有最小生成树的集合，

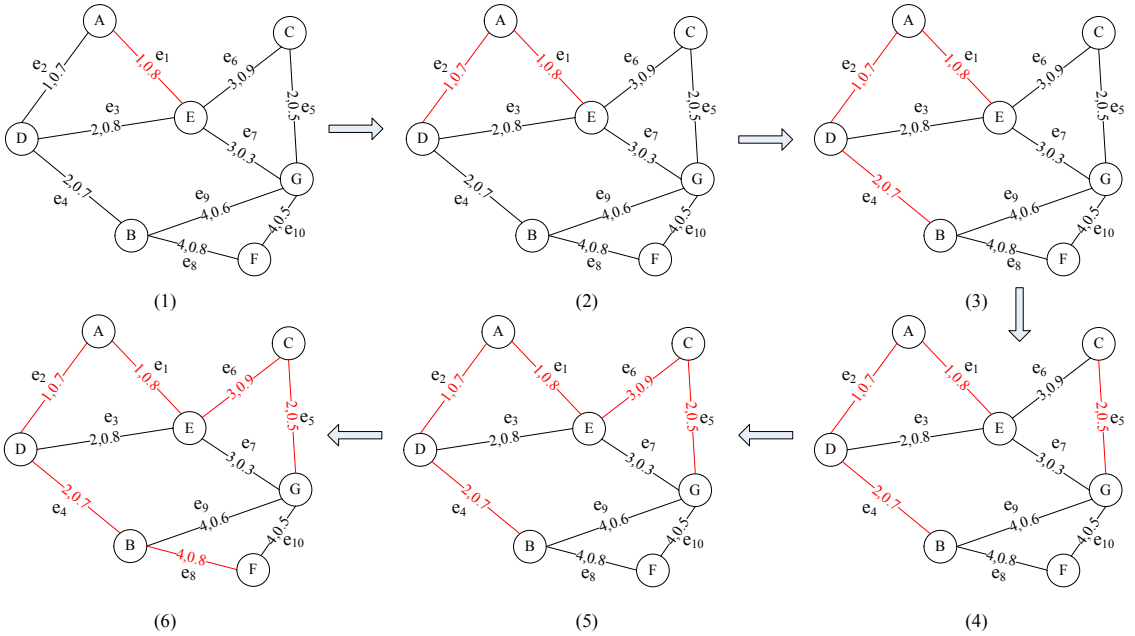

图 2.3 Example 2: 最优生成树计算过程。

若最小生成树T满足$\forall T' \in (Imp(\mathcal{T}) \setminus \{T_O^{\mathcal{G}}\})(R_{T'} \leq R_T)$，则称$T$为不确定图$\mathcal{G}$的次优生成树。记编号最小的次优生成树为$T_{SO}^{\mathcal{G}}$，其边集为$E_{SO}^{\mathcal{G}}$，边的权值和为$W_{SO}^{\mathcal{G}}$。

由定义 2.3.1可知，当最优生成树不唯一时，次优生成树也是最优生成树。显然，我们可以借用求次小生成树的思想求解次优生成树，步聚如下

1. 选择一条不在生成树$T_O^{\mathcal{G}}$中的边加入生成树$T_O^{\mathcal{G}}$；
2. 假设新加入的边为$e = (u, v)$，寻找生成树$T_O^{\mathcal{G}}$中从顶点u到顶点v的路径中优先级最高的边并删除。这样可以得到一颗新的生成树；
3. 重复前面2步，知道所有的边都已经处理，这样我们能够得到$|E| - |V| + 1$颗新的生成树，从这些生成树中求出最优的生成树则是不确定图$\mathcal{G}$的次优生成树。

在第二步中，我们需要知道最优生成树中任意两个顶点之间优先级最高的边，我们可以在求解最优生成树时进行预处理得到：

1. 设$F(i, j)$为在最优生成树中从顶点i到顶点j的路径中优先级最高的边；

2. 在求解最优生成树时，假设边$e=(u,v)$加入了边集$E_O^{\mathcal{G}}$，则对于图中所有顶点x，应该进行如下转换：

   (a) 如果边e比边$F(x,u)$优先级更高，则$F(x,u)=e$；

   (b) 如果边e比边$F(x,v)$优先级更高，则$F(x,v)=e$；

§2.4 本章小结

第三章　不确定图中Top-K生成树算法

§3.1　问题定义