



一种不确定图边发生故障最可靠最大流快速计算方法

李富豪

(东南大学 计算机科学与工程学院, 江苏 南京 211189)

摘要: 不确定性是很多现实应用系统的固有特性, 如智能电网中元件本身有存在的故障概率, 交通网络中运输线路的堵塞率等, 这些不确定性表达在图数据上形成不确定图。不确定图的最可靠最大流是指在可以传递最大流的多个流分布方案中寻找可靠性最大的方案, 研究该问题对于构建可靠性网络和分析系统薄弱环节等一系列问题, 具有重要的研究意义。在实际的网络中边有发生故障的概率, 因此, 当不确定图的边发生故障后, 如何快速的找到剩余不确定图的最可靠最大流分布也就变的很有意义。针对该问题, 本文首先设计了 BASE 算法, 该算法在不确定图的边发生故障之后, 通过状态划分算法重新计算剩余不确定图的最可靠最大流分布, 本文将 BASE 算法作为基础算法用于比较; 接着本文针对 BASE 算法重复计算复杂度较高的缺点, 提出了一种基于状态划分树的增量算法 ICA, 该算法的主要思想是保存不确定图在状态划分过程中的子图区间, 并构造子图区间树, 并通过子图区间判断边存在于子图的个数, 将不确定图的边分为 ABC 三类, 所以当边发生故障, 首先判断故障边类别, 对于 A 边通过割集对子图区间树剪枝获取满足最大流的区间, 对于 B 边, 使用 B-CESA 算法获取依然满足最大流的区间; 对于 C 边则不需要计算。最后本文通过实验, 比较了两种算法的性能。实验结果表明 ICA 算法相对于 BASE 算法其空间复杂度有一定的增加, 但是时间复杂度方面具有较大的优势。

关键词: 不确定图; 最大流; 分布可靠性; 边故障

A Method To Get Most Maximum Flow when Uncertain Graph Edge Break Down

Li Fuhao

(School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

Abstract: Uncertainty is the intrinsic characteristics of many systems, such as the failure probability of element which exists in the smart grid, transportation network, etc. The uncertain graph is formed when the expression of uncertainty in graph. The most reliable maximum flow of uncertain graph is the plan with maximum reliability which can communicate the multiple streams of maximum flow distribution scheme. To study the problems has important research significance for building network reliability and analysis system for weak links, and a series of problems. In the actual network, the edge in which has the probabilities of failure, therefore, when uncertain graph edge failure, how to quickly find the most reliable flow distribution of the residual uncertain graph also becomes very meaningful. To solve this problem, this paper designed the BASE algorithm. After failure occurs, use state partition algorithm to calculate the most reliable and maximum flow distribution of the residual uncertain graph. In this paper, the BASE algorithm is the basic algorithm to compare with other algorithm. Then aiming at the shortcomings of high repeat BASE algorithm computational complexity, this paper proposes a Increment Cost Algorithm with State Interregional Tree (ICA). The main ideas of the algorithm is to save the subgraph in the state of interval of the uncertain graph, and build the subgraph interval tree structure, and through the subgraph interval to judge the number that every single edge exist in the subgraph, uncertain graph edge can be divided into ABC three categories. Judge fault edge categories when edge failure occurs, the first category, for A edge then use cut set pairs figure tree pruning access to meet the maximum flow interval, for B edge, use B-CESA algorithm to obtain still meet the range of maximum flow; Do not need to compute for C edge. At the end of the paper, through the experiment compares the performance of the two algorithms. The experimental results show that the ICA algorithm has a little space complexity increases then BASE, but the time complexity has great advantage.

Keywords: Uncertain graph; maximum flow; flow reliability; edge failure

1 引言

现实世界中,网络形式的系统随处可见,例如,因特网、智能电网、科研合作网络、客户关系网络等。对于这些系统而言,不确定性是其固有特性,如智能电网中元件本身存在的故障概率,交通网络中运输线路的堵塞率等,当这些不确定性表达在图数据上就形成不确定图^[1-3]。

目前对于不确定图理论的研究有很多,很多有价值的问题及其处理方法近年来不断被提出^[1-13],首先,针对不确定子图,文献[1-3]就不确定图中如何挖掘最可靠子图进行了深入的研究,依次使用路径覆盖、启发式的贪心算法、蒙特卡洛仿真等算法解决该问题;邹兆年、张硕、韩蒙等人分别研究了不确定图上频繁子图挖掘、不确定图数据库查询处理等问题^[4-8],针对挖掘模型构建、降低子图同构的时间复杂度以及k极大频繁模式挖掘等问题,提出了有效的算法;而袁野等人^[9]使用过滤-验证的思想研究了子图相似性查询,利用概率矩阵索引最大化的过滤候选集,再通过取样算法验证相似性。其次,对于不确定图上的可达性、近邻查询等问题,文献[10, 11]分别提出了各自的k近邻的定义和高效的查询处理算法解决top-k近邻查询;张应龙等人则提出了基于SimRank度量的方法求解不确定图的k-NN查询;文献[13,14]在不确定图上拓展了最短路径问题,并给出了非常有参考意义的解决思路和算法。此外,随机流网络的可靠性问题是传统流问题在不确定图上的自然延伸,其定义是从源点到汇点传输额定流量的可靠性。该问题反映了网络负载流量的能力,是运筹学的经典问题,已被证明是NP-hard问题^[15,16]。文献[17-24]对该问题进行了深入的研究,提出了很多相关算法,包括针对小规模图的完全枚举状态空间算法、状态空间划分等精确算法,以及针对大规模图的蒙特卡洛仿真、机器学习等近似算法。不确定图的最可靠最大流是指在可以传递最大流的多个流分布方案中需找可靠性最大的方案,研究该问题对于构建可靠性网络和分析系统薄弱环节等一系列问题,具有重要意义^[25-27],因为在实际网络中边有发生故障的概率,因此,当不确定图的边发生故障后,如何快速的找到剩余不确定图的最可靠最大流分布也就变的很有意义。目前对于不确定图最可靠最大流分布的研究已经成熟

^[25-27],其中蔡伟等人在文献[25]中提出的基于简单路径组合思想提出了一种求解最可靠最大流的算法SPCA,通过简单路径流量的组合,在无需求得所有最大流分布的情况下获得最可靠最大流;接着针对SPCA算法易受路径数量及瓶颈容量影响的问题,又提出一种基于状态空间划分的最可靠最大流算法SDBA;杨娟等人^[26]出了基于概率和割集双过滤的状态空间划分算法DF-SDBA,相对于SDBA算法,DF-SDBA算法有效地减少了需要划分的区间,很大程度上克服了网络规模和稠密度对算法性能的影响;生衍等人^[27]提出一种基于具有普遍适用性的最可靠最大流解决方案,基于负权群落消去的NWCE算法、基于时间约束优先单环消去的SPEA-t算法和基于概率阈值约束优先单环消去的SPEA-p算法。

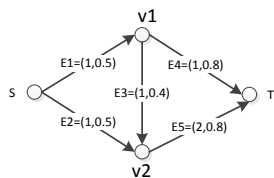
针对不确定图边发生故障,快速计算剩余不确定图的最可靠最大流分布的问题,本文首先设计了BASE算法,该算法在不确定图的边发生故障之后,通过状态划分方法^[25]重新计算剩余不确定图的最可靠最大流分布,本文将BASE算法作为基础算法用于比较;接着本文针对BASE算法重复计算复杂度较高的缺点,提出了一种基于状态划分树的增量算法ICA,该算法的主要思想是保存不确定图在状态划分过程中的子图区间,并构造子图区间树,并通过子图区间判断边存在于子图的个数,将不确定图的边分为ABC三类,对于A边移除之后,通过割集对子图区间树剪枝获取满足最大流的区间;对于B边移除,通过B-CESA算法获取依然满足最大流的区间,在剩余的子图中寻找满足最可靠最大流的子图;对于C类边移除不需要计算,依然是原有的最可靠最大流分布。所以当不确定图的边发生故障,首先判断故障边的类别,然后按照相应的方式进行处理即可。最后本文通过实验,比较了两种算法的性能。实验结果表明ICA算法相对于BASE算法其空间复杂度有一定的增加,但是时间复杂度方面具有较大的优势。

2 相关概念

本节对不确定图,不确定图s-t流,不确定图s-t流的可靠性等概念给出了相关的定义,为问题和算法的描述提供方便。

定义 1. (不确定图) 一个不确定图G是一个五元组 $G=(V, E, s, t, (C, P))$,其中,V是有

向图 G 中顶点的集合, E 是 G 中边的集合, s 和 t 分别为 G 的源点和汇点, (C, P) 是一个二元组且 $C: E \rightarrow N$ 是边上容量函数, $P: E \rightarrow (0, 1]$ 是边上的概率函数, 表明该边能通过的最大容量为 C 时的概率为 P , 当边不存在, 即边上能通过的容量为 0 时对应的概率为 $1-P$ 。


图 1 不确定图 G

例 1: 图 1 所示为一个不确定图 G , 该不确定图的源点为 s , 汇点为 t , 除此之外还包含其他的顶点 v_1 和 v_2 , 边集为 $\{E_1, E_2, E_3, E_4, E_5\}$, 以边 E_4 为例, 边上的容量 $c(E_4) = 1$, 概率 $p(E_4) = 0.8$, 也就是说 E_4 边能够达到流量为 1 的概率为 0.8, 而容量是 0 的概率为 $1 - p(E_4) = 0.2$ 。

定义 2. (剩余不确定图) 对于一个不确定图 G , 如果有一条边 e' 被移除, 剩下的不确定图被称为原不确定图 G 移除边 e' 之后的不确定图 G' , 则 G' 可以表示为 $G' = (V, E - e', s, t, (C, P) - e'(c, p))$, 其中剩余不确定图的顶点与原不确定一致, 且同时删除被移除边上的容量和概率的对应关系。

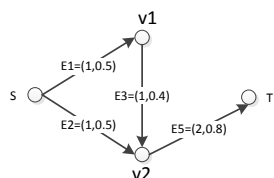

图 2 G 在移除边 E_4 后的剩余不确定图 G'

图 2 为不确定图 G 移除边 E_4 之后的剩余不确定图 G' , G' 中的顶点与原不确定图一致, 但是边以被移除边上的流量概率对应被移除。

定义 3. (不确定图的子图) 不确定图 $G = (V, E, s, t, (C, P))$ 的子图 $g = (V', E', s, t, C')$ 是一个确定图, 其中 $V' = V$, $E' \subseteq E$, C' 是容量的集合, 且 C' 满足 $\forall e \in E', C'(e) = C(e)$ 。如果 $E' = E$, 则称 g 为不确定图 G 的最大子图, 记作 $MSG(G)$ 。

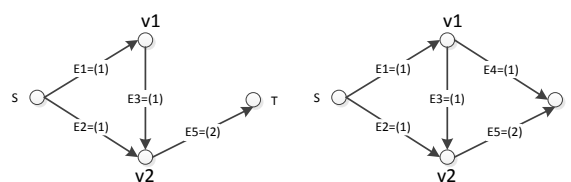

图 3 G 的子图 g_1

图 4 G 的子图 g_2

如图 3 和图 4 所示, g_1 和 g_2 是图 1 中不确定图 G 的两个子图, 根据定义子图 g_2 为不确定图 G 的最大子图, $MSG(G)$ 。

定义 4. (不确定图的子图概率) 不确定图 $G = (V, E, s, t, (C, P))$ 的一个子图 $g = (V', E', s, t, C')$, 则该子图 g 的概率为 $P(G) = \prod_{e \in E'} P_e * \prod_{e \notin E' \text{ 且 } e \in E} (1 - P_e)$ 。即子图中边的存在概率和不确定图中其余边的不存在概率之积。

例 2: 根据上述的子图概率计算的公式可得, 不确定图 G 的子图 g_1 的概率为 $P(g_1) = 0.5 * 0.4 * 0.5 * 0.8 * (1 - 0.8) = 0.016$, 同样的, 可以知道其最大子图 g_2 的子图概率为 $P(g_2) = 0.5 * 0.5 * 0.4 * 0.8 * 0.8 = 0.064$ 。

定义 5. (不确定图 s-t 流) 给定源点 s 和汇点 t , 不确定图 G 上的 s - t 流 f 是一个映射, 它把每条边对应到一个非负实数, $f: E \rightarrow R^+$; 值 $f(e)$ 直观上表示由边 e 携带的流量。一个流 f 必须满足下面两个性质:

(1) (容量条件) 对每条边 $e \in E$, 有 $0 \leq f(e) \leq c(e)$ 。

(2) (守恒条件) 除了 s 和 t 之外, 对每个结点, 有 $\sum_{e: e \rightarrow v} f(e) = \sum_{e: v \rightarrow e} f(e)$, 即: 所有进入结点的流值之和等于所有从该结点出来的流值之和。

定义 6. (不确定图最大流) 不确定图 G 中, 在每条边都存在时, 能够从源点 s 到汇点 t 传输的最大流值 f_{\max} 为不确定图 G 的最大流。

例 3: 图 1 所示的不确定图 G 能够达到的最大流为 2。

定义 7. (不确定图的 d 容量可靠性) 不确定图的 d 容量可靠性 $P(d)$ 可以表示为不确定图所有能够满足 d 流的子图概率之和^[25], 即 $P(d) = \sum_g \{p(g) | f(g) \geq d\}$, 特别的, 当 d 为不确定图的最大流时, 表示的为不确定图相对于最大流的容量可靠性。

例 4. 如图 5 为图 1 所示不确定图 G 的三个能

够满足最大流 2 的所有子图，那么不确定图相对于最大流的容量可靠性可以表示为 $P = p(g_1) + p(g_2) + p(g_3) = 0.176$ 。

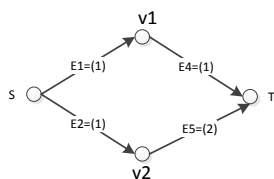


图 5 不确定图 G 够满足最大流 2 的子图 g_3

定义 8. (不确定图流分布及其可靠性) 给定一个不确定图 $G = (V, E, s, t, (C, P))$ 及其上的一个流分布 F ，则 F 的可靠性是不确定图 G 所有包含该流分布 F 的子图概率之和，可以表示为 $P(d) = \sum \{p(g) | g \geq F\}$ 同时也等于不确定图 G 中所有流量不为 0 的边的概率之积^[10]，即 $P(F) = \prod_{i=1}^{|E|} \{p(e_i) | f(e_i) > 0\}$ 。特别的，当流分布达到的流量为不确定图最大流时，为不确定图的最大流分布，对应的为不确定图的最大流分布可靠性。

例 5: 如图 6 为不确定图 G 的 2 个最大流分布 F_1 和 F_2 ，分析可知，最大流分布 F_1 被子图 g_2 和 g_3 所蕴含，则 $P(F_1) = P(g_2) + p(g_3) = 0.16$ ，此时也可以根据分布中流量不为 0 的概率之积来计算，此时的计算过程为 $P(F_1) = 0.5 * 0.5 * 0.8 * 0.8 = 0.16$ ，结果一致。同理，对于分布 F_2 ，通过定义可知 $P(F_2) = 0.08$ 。

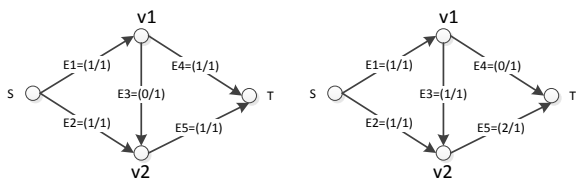


图 6 不确定图 G 的两个最大流分布

定义 9. (不确定图最可靠流分布) 一个不确定图最可靠流分布是一个不确定图流分布，其可靠性不小于该不确定图上的任一同等流分布。特别的，当流量为最大流时，为最可靠最大流分布，其对应的可靠性为最可靠最大流分布可靠性。

例 6. 如图 6 中的两个分布是不确定图 G 的两个最大流分布，其中 $P(F_1) > P(F_2)$ ，且不存在一个其他的最大流分布 F ，使得 $P(F) > P(F_1)$ ，则

F_1 是不确定图 G 的最可靠最大流分布，其对应的可靠性为不确定图 G 的最可靠最大流分布可靠性。

3 基于重复计算的基本算法

基于重复计算的基础算法 (BASE algorithm) 的思想是，首先获取故障边，然后获取对应的剩余不确定图，然后在剩余不确定图上重新计算最可靠最大流分布。本文将 BASE 算法作为计算剩余不确定图最可靠最大流分布的基本算法，用于对比，同时，为了更好的表示算法的性能，本文对于不确定图的所有边都假设移除一次，然后重新计算，最后获取平均值作为比较。

根据上述分析，下面给出重复计算的基础代码实现：

算法 1. BASE 算法

输入：不确定图 G ，源点 source，汇点 sink

输出：不确定图每一条边移除之后对应的最可靠最大流分布 PF 和其对应可靠性 P，算法消耗的平均时间

1. Input uncertain Graph G and its source point and sink point;
2. As for all edges in uncertain Graph G , get it's Remain uncertain Graph G' ;
3. Get G' maxFlow and PF and P
4. Get average time cost for every single edge
5. Return PF, P and average time cost.

BASE 算法本身就是一次关于最可靠最大流的计算，所以和最可靠最大流分布的算法复杂度有关，本文使用的计算方式是 SDBA 算法，其复杂度为 $O(k|V|^2|E|)$ ，其中 K 为划分过程中需要运行 Dinic 算法的次数。所以最后的 BASE 的算法复杂度也可以表示为 $O(k|V|^2|E|)$ 。

4 基于状态划分树的增量算法

本文提到的 BASE 算法只是作为基本算法作为比较，在图规模较大或者高稠密度图的情况下显然是不合适的。本节提出了一种基于状态划分树的增量算法 ICA (Increment Cost Algorithm with State Interregion Tree)，该算法的主要思想是保存不确定图在状态划分过程中的子图区间，并构造子图区间树，并通过子图区间判断边存在于子图的个数，将

不确定图的边分为 ABC 三类，然后分别做不同的处理，对于 A 边移除之后，通过割集对子图区间树剪枝获取满足最大流的区间；对于 B 边移除，通过 B-CESA 算法获取依然满足最大流的区间；对于 C 类边移除不需要计算，所以当不确定图的边发生故障，首先判断故障边的类别，然后按照相应的方式进行处理即可。

4.1 构建状态划分树

对于不确定图 G 而言，求解最可靠最大流分布的方式有很多，目前普遍使用的是算法 SDBA，SDBA 算法中使用到了状态划分算法，在状态划分的过程中，可以生成一个区间状态树，这样的一个区间状态树生成的步骤如下：

1. 将全集状态 $C[L, U]$ 压入栈 S ， $C[L, U]$ 作为树的根节点；
2. 当栈中还有元素，出栈获取一个元素 C ，获 C 上界，将 C 划分为 n 个区间；
3. 将这 n 个区间作为 C 的子节点，如果子节点满足最大流未知，则继续要入 S ，继续第二步；
4. 当 S 为空，返回这个树。

如图 7 所示，为图 1 中不确定图 G 的状态划分过程，其中斜体下划线的状态为未知状态，需要继续划分，加粗字体的子图区间为满足最大流的状态区间，而其他的区间为

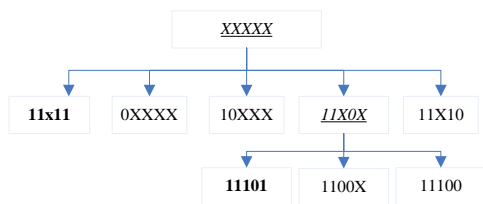


图 7 完全状态划分树 State-Tree

如图 7 所示，在完全状态划分树 State-Tree 的基础上，去掉不满足最大流的分支，则形成只满足最大流的状态划分树 MaxFlow-State-Tree，MaxFlow-State-Tree 的每一个分支的状态区间中的子图都满足最大流。

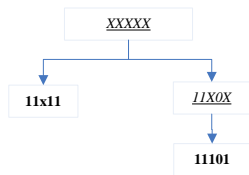


图 8 只满足最大流的状态划分树 MaxFlow-State-Tree

基于以上步骤，ICA 算法可以表示为如下 9 步骤：

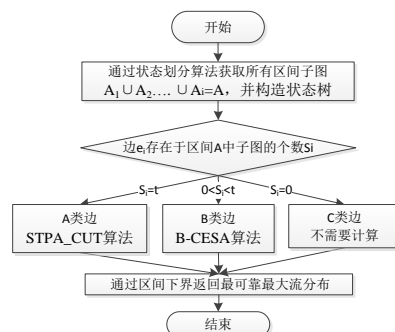


图 9 算法 ICA 流程图

4.2 不确定图边的定性分类

根据状态划分规则可以求解不确定图满足最大流的所有子图区间，本文根据边能够被满足最大流的子图的个数对边进行分类。根据不确定图子图的向量表示法以及向量的偏序关系，本文在获取的子图区间的基础上定义边在所有满足最大流子图的存在率 S_i 。

首先定义 s_{ij} 为边 e_i 在子图区间 j 的存在状况，根据偏序规则，其中 $e_i=1$ 为在子图区间 j 中的所有子图都必须包含边 e_i ； $e_i=0$ 表示在子图区间 j 中的所有子图都不包含边 e_i ；如果 $e_i=x$ 则表示子图区间 j 中的子图有包含和不包含两种情况，定义 s_{ij} 如下：

$$s_{ij} = \begin{cases} 1 & e_i=1 \\ 0 & e_i=x, 0 \end{cases} \quad \text{公式 1}$$

其中 i 表示边的序号， j 表示子图区间的序号。

对于不确定图中的每一条边 e_i 都有边存在率 S_i ，定义如下

$$S_i = \sum_{j=1}^{\tau} s_{ij} \quad \text{公式 2}$$

其中 i 为边的序号， $i \in \{1, \dots, n\}$ ， n 为不确定图 G 边的个数， j 为子图区间的序号， $j \in \{1, \dots, \tau\}$ ， τ 为使用状态划分后闭合区间的个数。

根据定义我们可知， S_i 有如下性质：

$$0 \leq S_i \leq \tau$$

其中 τ 为使用状态划分区间算法获得的最大流区间的个数。根据边存在率 S_i ，对边进行定性的分类如下：

1. $S_i = \tau$ ，那么 e_i 为 A 类边；

2. $0 < S_i < \tau$, 那么 e_i 为 B 类边;

3. $S_i = 0$, 那么 e_i 为 C 类边;

根据边存在率 S_i 的定义可知, 如果 $S_i = \tau$, 那么边 e_i 存在于所有的满足最大流的子图中; 如果 $S_i = 0$, 表示边 e_i 不存在任意一个满足最大流的子图中; 如果 $0 < S_i < \tau$, 表示边 e_i 存在于部分, 满足最大流的子图中。

根据以上的定性分析, 可以得到如下的一系列的推论。

性质 1. 对于不确定图 G 中的一条边 e_i , 如果边 e_i 是 A 类边, 那么移除边 e_i 会使得不确定图不能满足原有最大流。

证明: 想要证明移除 A 类边, 会使得不确定图 G 最大流减少, 需要证明两点 (1) A 类边移除, 不确定图 G 不能达到最大流; (2) 非 A 类边移除, 不确定图 G 任然可以达到最大流。下面将按照这两条分别证明;

(1) 因为 e_i 是 A 类边, 有 $S_i = \tau$, 根据定义边 e_i 存在于所有的满足最大流的不确定子图中, 假设经过划分算保存下来的所有闭合子图空间集为 $\{C_1, C_2, \dots, C_\tau\}$, 满足 $C_i \cap C_j = \emptyset$, $C_1 \cup C_2 \cup \dots \cup C_\tau = A$, 其中 τ 为满足最大流的子图区间的个数, 因为 $S_i = \tau$, 所以满足最大流的子图 g 一定有 $g \in A$ 且有 $e_i \in g$, 如果当 e_i 发生故障之后, 剩余不确定图 G' 的任意子图 g' 都不包含 e_i , 即 $e_i \notin g'$, 则必有 $g \notin A$, 因此对于 e_i 发生故障后的剩余不确定图 G' 的任意子图都不能达到最大流, 那么 G' 也不能达到最大流;

(2) 同理, 对于一条非 A 类边, 因为 $S_i < \tau$, 则在 A 中必存在一个子图 g , 使得 $e_i \notin g$, 如果边 e_i 发生故障, 不确定图可以使用子图 g 传递流量, 同时因为 $g \in A$, 即 g 能传递的最大流仍然为最大流, 所以对于非 A 类边发生故障, 不确定图 G 任然能够达到最大流;

的证。

例 7: 如图 1 中的不确定图 G , 经过状态划分过程, 获取的状态划分树如图 7 所示, 图 8 为其满足最大流的状态树, 其满足最大流的区间有两个, 分别是 $11x11$ 和 11101 , 因为经过划分有两个区间, 则 $\tau=2$, 根据公式 1, 可知对于边 e_1 , 有 $s_{11}=1, s_{12}=1$, 根据公式 2 可知, 对于边 e_1 的边存在率 $S_1=s_{11}+s_{12}=2=\tau$, 所以对于不确定图 G 而言, 边 e_1 是一个 A 类边, A 发生故障使得不确定图的最大流

从原来的 2 下降为 1。

性质 2. 对于不确定图中的一条边 e_i , 如果边 e_i 是 B 类边, 那么移除 e_i , 不确定图依然可以达到最大流。

证明: 对于不确定图 G 的任意一条边 e_i , 如果边 e_i 是 B 类边, 即如果有 $0 < S_i < \tau$, 则 e_i 移除之后, 可以找不包含边 e_i 的子图达到最大流, 使得最大流不会下降, 但是 e_i 的断掉使得一部分能够达到最大流的子图遭到破坏, 使得随机流网络的可靠性下降。

经过划分算保存下来的所有闭合子图空间集为 $\{C_1, C_2, \dots, C_\tau\}$, 满足 $C_i \cap C_j = \emptyset$, $C_1 \cup C_2 \cup \dots \cup C_\tau = A$, 其中 τ 为满足最大流的子图区间的个数, 因为 $0 < S_i < \tau$, 则对于任意的 $e_i \in g$ 子图, 可以将子图的集合 A 分为 $g \in A_1$ 和 $g \notin A_2$, 其中满足 $A_1 \cap A_2 = \emptyset$, $A_1 \cup A_2 = A$ 。

在 e_i 断掉之前, 随机流网络的可靠性可以为 A 中的所有子图概率之和, 可以表示为 $P(A)$, 而当 e_i 断掉之后, 集合 A_1 中的子图不一定能够满足最大流, 但是集合 A_2 中的流量可以满足最大流, 因此保证了不确定图的连通性不变; 同时 e_i 断掉使得 A_1 中至少有一个子图不能满足最大流, 因此使得随机流网络的可靠性下降, 此时即证明 e_i 为不确定图的可靠性关键边。

的证。

例 8: 如图 1 中的不确定图 G , 经过状态划分过程, 获取的满足最大流的区间分别是 $11x11$ 和 11101 , 其中 $\tau=2$, 根据公式 1, 可知对于边 e_3 , 有 $s_{31}=0, s_{32}=1$, 根据公式 2 可知, 对于边 e_3 的边存在率 $S_3=s_{31}+s_{32}=1 < \tau$, 所以对于不确定图而言 e_3 是一个 B 类边, 则 e_3 移除, 使得子图 11111 和 11101 不能满足最大流, 但是 11011 依然满足最大流, 此时的最大流可靠性下降, 仅为 11011 子图的可靠性。

性质 3. 对于不确定图中的一条边 e_i , 如果 e_i 是 C 类边, 那么移除边 e_i , 不确定图的最大流, 分布可靠性和容量可靠性都不发生改变。

证明: 对于不确定图 G 中的任意一条边 e_i , 如果 e_i 是 C 类边, 即如果 $S_i=0$, 则 e_i 断掉之后, 原先满足最大流的子图任然能够满足最大流, 使得不确定图的最大流不会改变, 即连通性不会改变, 随机流网络的可靠性也不会改变, 此时边 e_i 为不确定图的非关键边。

想要证明 $S_i=0$ 时, e_i 是不确定图 G 的非关键边, 只需要证明当 e_i 断掉之后, 不确定图能达到的最

大流和随机流网络可靠性不发生改变，即连通性和可靠性都不变即可。

经过划分算法保存下来的所有闭合子图空间集为 $\{C_1, C_2, \dots, C_\tau\}$ ，满足 $C_i \cap C_j = \emptyset$ ， $C_1 \cup C_2 \cup \dots \cup C_\tau = A$ ，其中 τ 为满足最大流的子图区间的个数，因为 $S_i=0$ ，那么对于所有的子图区间每一个对应的边 e_i 处都为标记“x”，即任意一个子图区间可以表示为 $C=(c_1, c_2 \dots x \dots c_n)$ ，其中 c_i 表示为子图区间C的各条边对应的表示， $i=\{1 \dots i-1, i+ \dots n\}$ ，则C可以分为两个子图区间(或子图) $C_1=(c_1, c_2 \dots 0 \dots c_n)$ 和 $C_2=(c_1, c_2 \dots 1 \dots c_n)$ ，当边 e_i 发生故障之后， $C_2=C_1$ ，因为 C_1 中的子图都能够满足最大流，故 e_i 断掉之后 C_2 中的子图也都能够满足最大流，所以不确定图的连通性和随机流可靠性都不会发生改变，此时 e_i 为不确定图G的非关键边。

的证。

例 9: 假设对于一个不确定图，其经过划分获取的子图区间只有一个，11x11，那么根据公式 1 和公式 2，边 e_3 为C类边，因此 e_3 移除之后，依然可以满足最大流。移除 e_3 之前，容量可靠性 $p_c=p(e_1)p(e_2)(1-p(e_3))p(e_4)p(e_5)+p(e_1)p(e_2)(e_3)p(e_4)p(e_5)=p(e_1)p(e_2)p(e_4)p(e_5)$ ，移除 e_3 之后，容量可靠性表示为 $p_c'=p(e_1)p(e_2)p(e_4)p(e_5)$ ，显然移除 e_3 之后不确定图的容量可靠性不变。

4.3 不确定图中边移除后定量计算

(1) A 类边的计算

对于A类边，由性质1可知，其断掉会使流量减少，即不能达到原有的最大流，此时原有最大流的网络可靠性和流分布的可靠性都为下降为0，理论上新的最大流相关的计算需要重新计算。但是可以根据该边对原有的完全状态划分树 State-Tree 进行处理，将原有的A边位置为1状态的去除，那么留下来的状态划分树的区间是移除A边之后的所有子图，在搜索子图区间树的过程中，使用割集中的边，对 State-Tree 剪枝，可以有效的减少待搜索区间个数。

例 10: 图 7 为不确定图 G 的完全状态划分树，对于不确定图 G 的 A 边 e_5 ，移除之后按照以上的处理方式，获取的状态树如下图 10 所示，所有的叶子节点为移除 e_5 之后的子图空间的划分。

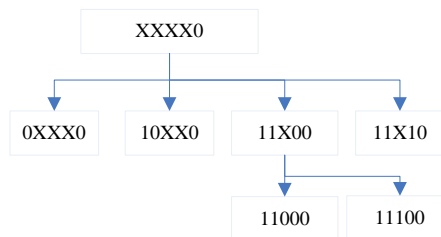


图 10 移除 e_5 之后的状态区间树

A类边根据搜索树的规则，对于每一个区间改边位置都设置为0，目的是获取不想交的区间。对于新构成的一个子图树，本文提出一种基于割集的状态树剪枝算法 STPA_CUT (State Tree Pruning Algorithm Based on Cut Set)，概算的主要是利用了割集中的边必定在最大流子图这一性质，通过割集中的边对于子图状态树进行剪枝，以达到减少搜索的目的。因为割集中的边一般在子图树的第二层就能达到剪枝的效果，所有该方法能够有效的减少搜索效果。

该算法首先获取根节点，对于其所有子节点，通过割集中的边进行剪枝，因为满足最大流的子图必定包含割集中的边，所以不包含割集中边的中间节点和叶子节点都必须舍去，对于依然保存的子图区间，进行进一步划分即可，即(1)当区间的下界子图满足最大流，则整个区间满足最大流，保留(2)当区间上界不满足最大流，则整个区间不满足最大流，舍弃(3)区间下界不满足最大流，上界满足最大流，则二次划分。通过这样的方式便可以有效的获取移除 e 之后所有满足最大流的区间子图，进而计算分布可靠性和容量可靠性。

例 11: 如图 11 是经过割集剪枝之后的子图树。其中有背景的圆部分及其子树都被剪枝掉。

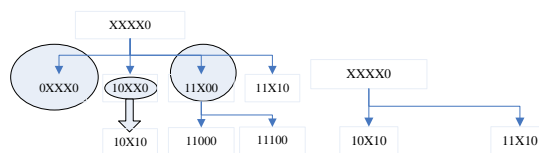


图 11 经过剪枝后的子图树

算法 2. STPA_CUT 算法

输入：割集边 CutSet，子图区间树 root，不确定图 G，源点 source，汇点 sink，移除边 e

输出：不确定图 G 在移除边 e 之后所有能够满足最大流的子图集合 set

1. get CutSet, root, init stack S;

2. push root in S;
3. if S is not empty, then pop S as C;
4. if the cut edge's position is 0 then move out C;
5. Else if C's lower bound graph satisfy Fmax then keep C
6. If C's upper bound graph not satisfy Fmax then move out C;
7. If C's lower bound graph not satisfy Fmax, but upper bound graph satisfy Fmax then divide C
8. Return all graph that satisfy Fmax after G move edge e

(2) B 类边的计算

对于 B 类边, 同样由性质 1 可以知道, 断掉 B 类边, 流量不会减少, 此时的重点在于计算剩余的不确定图的分布可靠性和容量可靠性。为了计算移除 B 边之后的分布可靠性和容量可靠性, 主要是计算移除 B 边, 原有只满足最大流的状态划分树 MaxFlow-State-Tree 中还剩哪些满足最大流的子图, 换言之, 就是要移除不能满足最大流子图, 且不能存在重复的子图。本文提出算法 B-CESA (B Class Edge Search Algorithm), 用于计算获取 B 边移除之后, 依然能够满足最大流的所有子图。该算法步骤如下:

算法 3. B-CESA 算法

输入: 满足最大流的状态划分树 MaxFlow-State-Tree, 移除边 e

输出: 移除 e 之后, 不确定图依然满足最大流的所有子图集合 set

1. For all leaf node in MaxFlow-State-Tree as C
2. If the e's position in C is 1, then move out C;
3. If the e's position in C is 0, then push C in set;
4. If the e's position in C is X, then divide C to $C_{e=1}$ and $C_{e=0}$ and goto step (2) (3);
5. Until MaxFlow-State-Tree is all be searched, then return set.

证明: 对于不确定图 G, 经过划分后获取的满足最大流的所有区间为 $\{C_1, C_2, \dots, C_\tau\}$, 满足

$C_i \cap C_j = \emptyset, C_1 \cup C_2 \cup \dots \cup C_\tau = A$, 其中 τ 为满足最大流的子图区间个数, 假设边 e_i 为 G 的 B 类边, 根据 B 类边定义, 存在一个 $k \in \{1, 2, \dots, \tau\}$, 使得 $\{C_1 \cup C_2 \dots \cup C_k\} = C_{e_i=0}$ 与 $\{C_{k+1} \cup C_{k+2} \dots \cup C_\tau\} = C_{e_i=1}$, 令 $C' = \{(C_1 - e_i) \cup (C_2 - e_i) \dots \cup$

$(C_k - e_i)\}$, 因为 $C_{e_i=0}$ 中的所有子图 e_i 位置为 0,

所以有 $C' = C_{e_i=0}$

若要证明 C' 为 $G - e_i$ 之后, 所有满足最大流的子图集合, 需要证明以下两点:

(1) $\forall g \in C'$, 都有 $f(g) = f_{max}$;

(2) $\forall g \in C_{e_i=1}$, 若 $f(g - e_i) = f_{max}$, 则必存在一个 $g_0 \in C'$, 使得 $g_0 = g - e_i$ 。

首先证明 (1): 已知条件 $\forall g \in C_{e_i=0}$ 都有 $f(g) = f_{max}$, 且 $C' = C_{e_i=0}$, 所以可以推出对于任意的子图 g , 如果 $g \in C'$, 都有 g 满足最大流, 即 $\forall g \in C'$, 都有 $f(g) = f_{max}$ 。

然后证明 (2): 反证法。假设 $\forall g \in C_{e_i=1}$, 若有 $f(g - e_i) = f_{max}$, 则对于 $\forall g_0 \in C'$, 都有 $g - e_i \neq g_0$ 。

对于 $g \in C_{e_i=1}$, $g - e_i = \{e_1, \dots, e_{i-1}, 0, e_{i+1}, \dots, e_{|E|}\}$ 且有 $f(g - e_i) = f_{max}$, 所以 $g - e_i \in C_{e_i=0}$, 又因为 $C' = C_{e_i=0}$, 所以 $g - e_i \in C'$, 这与不存在一个 $g_0 \in C'$ 使得 $g - e_i = g_0$ 矛盾。假设不成立。

的证。

例 11: 如图 1 中的不确定图 G, 经过状态划分获取满足最大流的子图区间有两个, 分别是 11x11 和 11101, e_3 是一个 B 边, 根据以上规则, 11x11 区间在 e_3 位置上的状态是 x, 可以分为 11111 和 11011 两个区间, 因为 11111 和 11101 两个区间在 e_3 位置上的状态是 1, 所以舍去, 而只保留区间 11011。所以最终移除 e_3 之后, 依然满足最大流的子图区间是 11011。

(3) C 类边的计算

对于 C 类边的计算, 根据性质 1, 其断掉不会影响连通性, 根据性质 3, 其断掉不会影响可靠性, 也就是说 C 类边断掉不会对不确定图的状态造成任何的影响。

根据性质 3, C 类边移除之后不需要计算, 直接使用原有的最大流, 分布可靠性和容量可靠性。

基于以上处理方式, ICA 算法的伪代码如下:

算法 4. ICA 算法

输入：不确定图 G ，源点 source，汇点 sink

输出：不确定图每一条边移除之后对应的最可靠最大流分布 PF 和其对应可靠性 P ，算法消耗的平均时间

1. Input uncertain Graph G and its source point and sink point;
2. Get maxFlow, StateMtrix and build StateTree using algorithm SDBA
3. Classify ABC edge by StateMtrix;
4. When edge e_i in G being moved, then check Edge class;
5. If e_i is A class edge then use algorithm STPA_CUT to get G' maxFlow PF and P ;
6. If e_i is B class edge then use algorithm B-CESA to get G' maxFlow PF and P ;
7. If e_i is C class edge then G' maxFlow PF and P is same as before e_i be moved.
8. Get average time cost for every single edge;
9. Return PF, P and average time cost.

4.4 算法复杂度分析

基于状态划分树的增量算法的主要思想是将不确定图的边分为 ABC 三类边，然后根据三类边的不同性质对于不同性质的边选择不同的算法，从而简化计算过程。

首先对于 A 类边，考虑最坏的情况，就是重复完全计算，该算法运算复杂度主要体现在 K 次运行 Dinic 算法，因此对于 A 边的复杂度为 $O(ak|V|^2|E|)$ ，其中 a 为不确定图 G 中 a 边的个数，对于 B 类边，需要遍历满足最大流的所有区间，其复杂度为 $O(b\tau)$ ，其中 b 为不确定图 B 边的个数， τ 为满足最大流区间的个数，对于 C 边来说，因为不需要计算，所以复杂度为 $O(c)$ ，其中 c 为不确定图 C 类边的个数。综上所述，基于状态划分树的增量算法的整体复杂度是 $(O(ak|V|^2|E|) + O(b\tau) + O(c)) / |E|$ ，其中 $a+b+c=|E|$ 。由公式可见，ICA 在最坏的情况下，复杂度和 BASE 算法的复杂度一致。

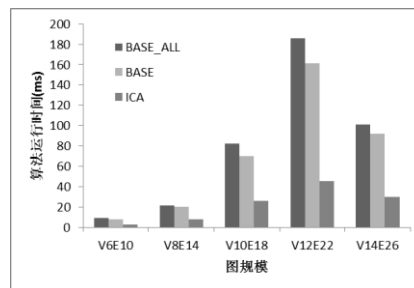
5 实验与分析

为了验证本文所提出算法的运行效率及分析影响算法性能各种因素，本文进行了一系列的实验，

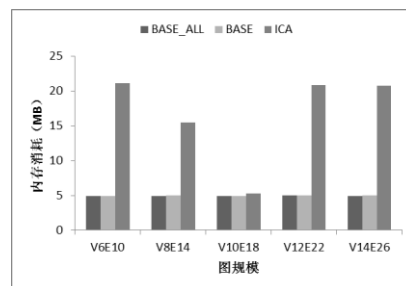
实验平台为一台 Intel Core 的 PC 机 (CPU i7-3770, 3.40GHz, 内存 8GB, 64 位 windows 7 操作系统)，算法采用 C++ 在 VS2010 上实现。

本文采用了与文献^[10]相同的数据集，使用 NETGEN 生成器^[11]产生 V_6E_{10} 、 V_8E_{14} 、 $V_{10}E_{18}$ 、 $V_{12}E_{22}$ 、 $V_{14}E_{26}$ 共 5 组不同图规模的二态有向图集合 (本文实验数据集大小为 5)，其中 V_nE_m 表示有 n 个顶点、 m 条边组成的图，图中边的容量与对应概率满足均匀分布。通过在不同图规模的情况下，比较算法 BASE 和 ICA 算法在运行时间及内存消耗方面的差异。实验中的 BASE_ALL 算法指代的是完全重复计算的方式。

实验 1. 不同图规模对于算法性能的影响。



(a) 时间消耗



(b) 内存消耗

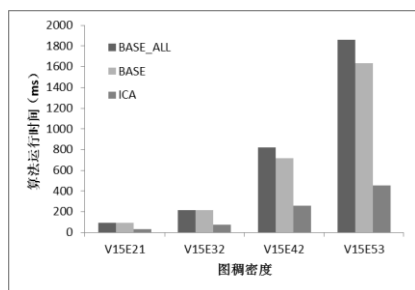
图 (12) 图规模对 BASE 和 ICA 时间和内存消耗影响

根据图 12 显示，算法 BASE_ALL 算法是完全重复计算的方式，BASE 算法是在当流量一致的情况下才计算分布可靠性和容量可靠性，如图所示，BASE 算法相对于 BASE_ALL 算法在时间上有一定的减少，但是依然不适应大规模图。如图 12 算法 ICA 相对于 BASE 算法在空间复杂度上有一定的增加，但是时间复杂度上有较大的优势。

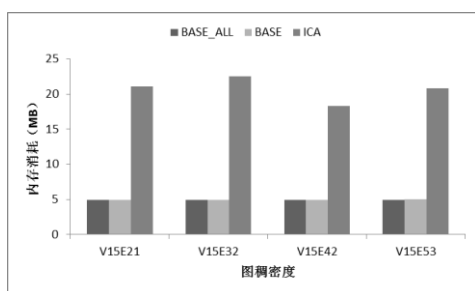
实验 2. 不同图稠密度对于算法性能的影响

为更好的反应算法 BASE 算法和 ICA 算法的性能差异，使用 NETGEN 生成器生成 E10D3、E16D3、E20D3、E24D3 四种不同稠密度的图，通过在不同图稠密度的情况下，比较算法 BASE 和 ICA 算法在

运行时间及内存消耗放方面的差异。实验中的 BASE_ALL 算法指代的是完全重复计算的方式。本节使用生成的 V_{15E21} , V_{15E32} , V_{15E42} , V_{15E53} 不同稠密的五组图, 实验结果如下:



(a) 时间消耗



(b) 内存消耗

图 (13) 图稠密度对 BASE 和 ICA 时间和内存消耗影响

如图 13 所示, 相对于不同稠密度的图, ICA 算法相对于 BASE 算法在空间复杂度上有一定的增加, 但是时间复杂度上有较大的优势。

6 结束语

针对不确定图边发生故障, 快速计算剩余不确定图的最可靠最大流分布的问题, 本文首先设计了 BASE 算法, 接着本文针对 BASE 算法重复计算复杂度较高的缺点, 提出了一种基于状态划分树的增量算法 ICA, 将不确定图的边分为 ABC 三类, 当不确定图的边发生故障, 首先判断故障边的类别, 然后按照相应的方式进行处理。实验结果表明 ICA 算法相对于 BASE 算法其空间复杂度有一定的增加, 但是时间复杂度方面具有较大的优势。本文的下一步工作, 一方面希望把这个衡量指标从最大流扩展到 d 流方面。

References:

[1] Hintanen P, Toivonen H and Sevon P. Fast discovery of reliable subnetworks//Proceedings of the 11th

International Conference on Advances in Social Networks Analysis and Mining (ASONAM). 2010: 104-111.

[2] Hintsanen P. The most reliable subgraph problem. //Proceedings of the 11th European Conference on Principles and Practice of Know-ledge Discovery in Databases. Warsaw, 2007: 471-478.

[3] Hintsanen P, Toivonen H. Finding reliable subgraphs from large probabilistic graphs. Data Mining and Knowledge Discovery, 2008, 17 (1) : 3-23.

[4] Zou ZN, Li JZ, Gao H, Zhang S. Mining frequent subgraph patterns from uncertain graphs. Journal of Software, 2009, 20 (11) : 2965-2976 (in Chinese with English abstract) . <http://www.jos.org.cn/1000-9825/3473.htm>

[5] Zou ZN, Li JZ, Gao H, Zhang S. Finding top-k maximal cliques in an uncertain graph. //Proceedings of the IEEE 26th International Conference on Data Engineering (ICDE), 2010: 649-652.

[6] Zou ZN, Li JZ, Gao H, Zhang S. Mining frequent subgraph patterns from uncertain graph data. IEEE Transactions on Knowledge and Data Engineering, 2010, 22 (9) :1203-1218.

[7] Han M, Zhang W, Li JZ. RANKING: An efficient K-maximal frequent pattern mining algorithm on uncertain graph database. Chinese Journal of Computers, 2010, 33 (8) : 1387-1395 (in Chinese) .

[8] Zhang S, Gao H, Li JZ, Zou ZN. Efficient query processing on uncertain graph database. Chinese Journal of Computers, 2009, 32 (10) : 2066-2079 (in Chinese with English abstract)

[9] Yuan Y, Wang GR, Chen L, Wang HX. Efficient subgraph similarity search on large probabilistic graph databases. In Proc. of VLDB, 2012: 800-811.

[10] Zhang HJ, Jiang SZ, Zou ZN. An efficient algorithm for top-k proximity query on uncertain graphs. Chinese Journal of Computers, 2011, 34 (10) : 1855-1896 (in Chinese with English abstract) .

[11] Zhang X, He XN, Jin CQ, Zhou AY. Processing k-nearest neighbors query over uncertain graphs. Journal of Computer Research and Development, 2011, 48 (10) : 1871-1878 (in Chinese with English abstract) .

[12] Zhang YL, Li CP, Chen H, Du LX. K-nearest neighbors in uncertain graph. Journal of Computer Research and Development, 2011, 48 (10) : 1850-1858 (in Chinese with English abstract) .

[13] D. D. M. L. Rasteiro, A. J. B. Anjo. Optimal paths in probabilistic networks. Journal of Mathematical Sciences, 2004, 120 (1) : 974-987.

[14] Li MP, Zou ZN, Gao H, Zhao ZL. Computing expected shortest distance in uncertain graphs. Journal of Computer



- Research and Development, 2012, 49 (10) : 2208-2220 (in Chinese with English abstract) .
- [15] Doulliez P, Jamouille E. Transportation networks with random arc capacities. *RAIRO*, 1972, 6 (3) : 45-49.
- [16] Alexopoulos C. Note on state-space decomposition methods for analyzing stochastic flow network. *IEEE Transactions on Reliability*, 1995, 44 (2) : 354-357.
- [17] A. Aven. Reliability evaluation of multistate systems with multi-state components. *IEEE Trans. Reliability*, 1985, 34: 473-479.
- [18] C. -C. Jane, J. -S. Lin, and J. Yuan. On reliability evaluation of a capacitated-flow network in terms of minimal cutsets. *IEEE Transactions on Reliability*, 1993, 42 (3) : 354-361.
- [19] J. -S. Lin, C. -C. Jane, and J. Yuan. On reliability evaluation of a capacitated-flow network in terms of minimal path sets. *Networks*, 1995, 25 (3) : 131-138.
- [20] S. Patra and B. Misra, Evaluation of probability mass function of flow in a communication network considering a multistate model of network links. *Microelectron Reliability*, 1996, 36: 415-421.
- [21] Chin-Chia Jane and Yih-Wenn Lai, A practical algorithm for computing multi-state two-terminal reliability. *IEEE Trans. Reliability*, 2008, 57: 295-302.
- [22] Chin-Chia Jane and Yih-Wenn Lai, Computing multi-state two-terminal reliability through critical arc states that interrupt demand. *IEEE Trans. Reliability*, 2010, 59: 338-345.
- [23] Ramirez-Marquez J E, Coit D W. A monte-carlo simulation approach for approximating multi-state two terminals reliability. *Reliability Engineering & System Safety*, 2005, 87: 253-264.
- [24] Rocco C M, Muselli S M. Approximate multi-state reliability expressions using a new machine learning technique. *Reliability Engineering & System Safety*, 2005, 89 (3) : 261-270.
- [25] C W, Zhang BL, Lv JH. Algorithms of the most reliable maximum flow on uncertain graph. *Chinese Journal of Computers*, 2012, 35 (11) :2371-2380 (in Chinese with English abstract) .
- [10] 张海杰, 姜守旭, 邹兆年. 不确定图上的高效 top-k 近邻查询处理算法. *计算机学报*, 2011, 34 (10) : 1885-1896.
- [11] 张旭, 何向南, 金澈清, 周傲英. 面向不确定图的 k 最近邻查询. *计算机研究与发展*, 2011, 48 (10) : 1871-1878.
- [12] 张应龙, 李翠平, 陈红, 杜凌霞. 不确定图上的 kNN 查询处理. *计算机研究与发展*, 2011, 48 (10): 1850-1858.
- [14] 李鸣鹏, 邹兆年, 高宏, 赵正理. 不确定图上期望最短距离的计算. *计算机研究与发展*, 2012, 49 (10) : 2208-2220.
- [25] 蔡伟, 张柏礼, 吕建华. 不确定图最可靠最大流算法研究. *计算机学报*, 2012, 35 (11) : 2371-2380.
- [26] 张柏礼, 杨娟, 吕建华, 等. 基于不确定图的最可靠最大流的改进算法[J]. *东南大学学报: 自然科学版*, 2015, 45 (2) : 241-246.
- [27] 张柏礼, 吕建华, 生衍, 等. 一种不确定图中最可靠最大流问题的解决方案[J]. *计算机学报*, 2014, 10: 003.

附中文参考文献:

- [4] 邹兆年, 李建中, 高宏, 张硕. 从不确定图中挖掘频繁子图模式. *软件学报*, 2009, 20 (11) :2965-2976. <http://www.jos.org.cn/1000-9825/3473.htm>
- [7] 韩蒙, 张炜, 李建中. RAKING:一种高效的不确定图 K-极大频繁模式挖掘算法. *计算机学报*, 2010, 22 (8) : 1387-1395.
- [8] 张硕, 高宏, 李建中, 邹兆年. 不确定图数据库中高效查询处理. *计算机学报*, 2009, 32 (10) : 2066-2079