基于流量和可靠性的不确定图关键边评估方法研究

张柏礼1,2, 李富豪1

1(东南大学 计算机科学与工程学院, 江苏 南京 211189)

2(计算机网络和信息集成教育部重点实验室(东南大学), 江苏 南京 211189)

摘 要: 网络中边的重要性评估是研究网络的一项重要内容, 现有网络中边的重要性评估方法有很多, 有的使用与边相关联节点的个数来衡量边的关键程度, 有的使用边介数(任意结点经过边的次数)大小反映边对于网络传输能力和控制能力强弱, 然而在不确定图当中, 其最可靠最大流的分布及其分布可靠性是我们关注的重点, 现有的评估方式基本没有涉及到分布可靠性, 因此不适合应用于以不确定图为背景的相关领域. 本文构建了基于流量和可靠性(分布可靠性和容量可靠性)指标的不确定图关键边评估的数学模型, 该模型针对不确定图中边移除(故障)后对流量和可靠性产生的相对损失这一角度, 对边的关键度进行综合评估, 该模型认为流量作为衡量关键边的最重要因素, 当流量一致时, 比较移除边之后的最大流分布可靠性, 为了增加区分度, 同时考虑容量可靠性. 对于边移除之后, 不确定图流量和可靠性相对变化的计算, 本文首先设计了BASE算法, 即通过比较不确定图在移除边之后能够达到的最大流, 只有当流量相同的情况下, 才计算比较分布可靠性和容量可靠性, 本文将BASE算法作为衡量关键边的基本算法; 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, 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, 通过子图区间判断边存在于子图的个数, 将边分为ABC三类, 当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; 当B边移除, 通过B-CESA算法获取依然满足最大流的区间; 当C类边移除不需要计算. 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, 关键边, 最大流, 分布可靠性, 容量可靠性

# 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. 如何在复杂网络环境下, 保证网络的可靠性与抗毁性[1, 2], 以及网络中边的重要性评估[3-5]是研究网络的重要内容. 由节点重要度评估找出那些重要的关键节点, 可以通过重点保护这些关键节点提高整个网络的可靠性. 因此, 对于网络中节点的重要度进行评估是一项有意义的工作.

评估网络中节点重要性的方法很多, 最简单的方法是以节点的连接度(节点连接的边数) 作为节点重要度的衡量标准[6], 认为与节点相连的边越多则该节点越重要. 这种评估方法具有片面性, 有些重要的”关键节点” 并不一定具有较大的连接度, 比如只有两条边相连的”桥节点”. 2002年Girvan和Newman在介数中心度的基础上, 提出了边介数[7](edge-betweenness)概念, 通过计算网络中边介数的大小来反映边对网络资源的传输能力和控制能力的强弱, 边介数越大, 表明网络中任意节点对经过该条边的次数就越多, 对网络资源的传输能力和控制能力就越强, 在网络中所起到的作用也就越大, 因此, 边介数在一定程度上反映了边的重要程度; 文献[8]提出了一种基于生成树数目的节点删除法, 定义最重要的节点为去掉该节点使得生成树数目最小的节点. 节点删除法的问题是如果多个节点的删除都使得网络不连通, 那么这些节点的重要度将是一致的, 从而使得评估结果不精确. 针对已有复杂网络节点重要性评估方法中片面强调节点的度而忽略了边对与之相连节点的支撑作用的缺陷, 熊金石等人提出了节点度和边介数共同作用下的评估数学模型[9], 以体现边对其端节点的支撑作用.

然而在不确定图当中, 其最可靠最大流的分布及其分布可靠性是我们关注的重点[10, 14-15], 现有的评估方式基本没有涉及到分布可靠性, 不适合应用于以不确定图为背景的相关领域. 为此, 本文构建了基于流量和可靠性指标的不确定图关键边评估的数学模型. 该模型针对不确定图中边移除(故障)后对流量和可靠性产生的相对损失这一角度, 对边的关键度进行综合评估. 该模型认为流量作为衡量关键边的最关键因素, 当流量一致时, 比较移除边之后的最大流分布可靠性, 为了增加区分度, 同时考虑容量可靠性. 接着, 本文在该模型的基础上, 设计了量一种基于重复计算的基本算法BASE(BASE algorithm), 该算法的主要思想是首先计算不确定图每一条边移除之后能够满足的最大流, 然后通过比较不确定图在移除边之后能够达到的最大流, 只有当流量相同的情况下, 才计算比较分布可靠性和容量可靠性, 本文将BASE算法作为衡量关键边的基本算法; 针对BASE算法重复计算复杂度较高的缺点, 本文提出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, 通过子图区间判断边存在于子图的个数, 将边分为ABC三类, 当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; 当B边移除, 通过B-CESA算法获取依然满足最大流的区间, 从而进一步获取对应分布可靠性和容量可靠性; 当C类边移除不需要计算. 最后本文通过实验, 比较了两种算法的性能, 并分析了影响算法性能的因素.

# 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->N是边上容量函数, P: E->(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v1和v2, 边集为{E1, E2, E3, E4, E5}, 以边E4为例, 边上的容量c(E4)=1, 概率p(E4)=0. 8, 也就是说E4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(E4)=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’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和原不确定一致, 且同时删除被移除边上的容量和概率的对应关系.



图2 不确定图G在移除边E4后的剩余不确定图G’

图2为不确定图G移除边E4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’∈E, C’是容量的集合, 且C’满足∀e∈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 不确定图G的子图g1 图4 不确定图G的子图g2

如图3和图4所示, g1和g2是图1中不确定图G的两个子图, 根据定义子图g2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))的一个子图为g(V’, E’, s, t , C’), 则该子图g的概率为. 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g1的概率为P(g1)=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g2的子图概率为P(g2)=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→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∈E, 有0≤f(e)≤c(e).

(2)(守恒条件)除了s和t之外, 对每个结点 , 有, 即: 所有进入结点 的流值之和等于所有从 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fmax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8. (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[10], 即, 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(g1)+ p(g2)+ p(g3)=0. 176.



图5 不确定图G的三个能够满足最大流2的所有子图g1, g2, g3

**定义9. (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 同时也等于不确定图G中所有流量不为0的边的概率之的边的概率之积[10], 即. 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F1和F2, 分析可知, 最大流分布F1被子图g2和g3所蕴含, 则P(F1)=P(g2)+p(g3)=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(F1)=0. 5\*0. 5\*0. 8\*0. 8=0. 16, 结果一直. 同理, 对于分布F2, 通过定义可知P(F2)=0. 08.

 

图6 不确定图G的两个最大流分布F1和F2

**定义10. (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(F1)>P(F2) , 且不存在一个其他的最大流分布F, 使得P(F)>P(F1), 则F1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的最可靠性最大流分布可靠性.

# 边的关键度模型

对于一个网络能够达到的最大流的大小, 体现出了网络的联通能力和传输能力, 这是一个网络的最基础应用, 例如, 网络的最大流受限于网络的最小割集, 也就是说最小割集中的任意一条边断掉都会使得真个网络可以传输的最大流下降, 这样的边就成了整个网络相对于流量的关键环节. 在一个以不确定图为背景的应用中, 分布可靠性体现出对应分布的稳定性和健壮性, 这些性质对于以不确定图为背景的相关应用都是至关重要的, 在不确定图中, 相对于分布可靠性的关键边对网络信息传输的可靠性起到关键作用, 这样的环节一但发生故障, 会使得整个网络的可靠性下降. 基于此, 本文构建了基于流量和分布可靠性指标的不确定图关键边评估的数学模型, 该模型针对不确定图中边移除(故障)后对流量和分布可靠性产生的相对损失这一角度, 对边的关键度进行综合评估, 该模型认为流量作为衡量关键边的最重要因素, 当流量一致时, 比较移除边之后的最大流分布可靠性, 为了增加区分度, 同时考虑容量可靠性. 这是一种按照因子关键程度分层排序的方式, 将重要的目标排在前面, 例如已知排成e1 (p, pd, pc), e2 (p, pd, pc), …, e|E| (p, pd, pc). 然后对第1个目标求最优, 找出所有最优解集合, 用R1表示, 接着在集合R1范围内求第2个目标的最优解, 并将这时的最优解集合用R2表示, 依此类推, 直到求出第K个目标的最优解为止. 将上述过程用数学语言描述, 即

其中, R为边的全集. F, Pd, Pc分别为不确定图G在移除边e之后, 剩余不确定图G‘能够达到相对的最大流, 及其对应的流分布可靠性和容量可靠性

这种方法有解的前提是R1, R2, R3等集合非空, 是在解决实际问题中, 一般情况有一个损失阈值, 本文另外给出一种允许宽容的方法. 所谓“宽容”是指, 当求解后一目标最优时, 不必要求前一目标也达到严格最优, 而是在一个对最优解有宽容的集合中寻找. 这样的模型可以用数学语言描述为:

其中, R为边的全集, 而ai, bi, ci>0是一个宽容限度, 可以事前给定.

在研究该问题中, 为简化模型, 本文使用的是二态网络, 即边的状态只有两种, 且只考虑单故障情形, 即只考虑一条边发生故障或断掉的情形, 而不存在组合边发生故障的情形. 其中边发生故障对不确定图造成的影响都是针对不确定图的最大流.

# 关键边计算方法

基于本文第三部分提出的不确定图关键边的衡量模型, 计算不确定图边的关键程度主要体现在, 当边移除之后, 剩余不确定图能够达到的最大流, 分布可靠性和容量可靠性的计算. 本文首先设计了重复计算的基础算法(BASE algorithm), 然后针对BASE算法重复计算复杂度高的缺点, 提出一种基于状态划分树的增量算法ICA(Increment Cost Algorithm with State Interregion Tree).

## 基于重复计算的基础算法

基于重复计算的基础算法(BASE algorithm)的思想是, 首先计算不确定图每一条边移除之后能够满足的最大流, 然后通过比较不确定图在移除边之后能够达到的最大流, 只有当流量相同的情况下, 才计算比较分布可靠性和容量可靠性, 本文将BASE算法作为衡量关键边的基本算法

根据上述分析, 下面给出重复计算的基础代码实现(伪代码):

**算法1**. BASE算法

输入: 不确定图G, 源点source, 汇点sink

输出: 根据关键程度排序好的keyEdgeSet, 以及移除各边之后不确定图对应的最大流, 分布可靠新和容量可靠性

1. Input uncertain Graph G and its source point and sink point;
2. Get Fmax = Dinic(G, source, sink);
3. Get G maxflow when every single edge being cut off ALLEdgeFlowDecrease[|E|];
4. Sort keyedge by ALLEdgeFlowDecrease[|E|];
5. When the flow (G-ei) equles the flow (G-ej) then;
6. reCompute the (G-ei)’s maxFlow, pd and pc using SDBA[10];
7. Soet KeyEdge by pd;
8. Return sorted key\_edge\_set;

算法1首先要使用getALLEdgeFlowDecrease函数获取所有边发生故障之后能够满足的最大流, 然后使用soetKeyEdge\_by\_flow函数对计算的流量进行排序, 当前后两条边断掉之后获得的流量不相等时, 根据模型定义, 直接就可以比较出两条边的关键程度, 而只有当两条边断掉获得流量相等的情况下, 无法比较的情形, 才会重复计算不确定图的分布可靠性和容量可靠性, 使用分布可靠性和容量可靠性来区分这两条边的关键程度.

基于重复计算的基础算法的运行效率与重复计算的次数紧密相连, 在算法的首先需要计算所有边发生故障之后的最大流, 本文使用的计算最大流的算法是Dinic算法, 运行Dinic算法所需的处理时间为O(|V|2|E|), 因此在处理这部分是复杂度是O(|V|2|E|2); 然后当移除边之后流量相同的情况下, 需要重复计算, 本文使用的重复计算方式为基于状态划分的算法[10], 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. 因此整个BASE算法的复杂度为O(km|V|2|E|) + O(|V|2|E|2), 其中m为需要重复计算的次数, 一般m的范围是0≤m≤|E|.

## 基于状态划分树的增量算法

虽然BASE算法相对于完全重复计算的方式减少了不少的重复计算, 但是如果图规模较大的情况下, 任然需要多次的重复计算, 这显然对于大规模图或者高稠密图是不使用的. 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, 通过子图区间判断边存在于子图的个数, 将边分为ABC三类, 根据三类边不同的性质, 对三类边进行分别计算, 当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; 当B边移除, 通过B-CESA算法获取依然满足最大流的区间; 当C类边移除不需要计算.

### 构造状态划分树

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

### 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Si.

首先定义sij为边ei在子图区间j的存在状况, 根据偏序规则, 其中ei=1为在子图区间j中的所有子图都必须包含边ei; ei=0表示在子图区间j中的所有子图都不包含边ei; 如果ei=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sij如下:

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ei都有边存在率Si, 定义如下

公式2

其中i为边的序号, i∈{1…. n}, n为不确定图G边的个数, j为子图区间的序号, j∈{1…*τ*}, *τ* 为使用状态划分后闭合区间的个数.

根据定义我们可知, Si有如下性质:

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Si, 对边进行定性的分类如下:

1、Si=*τ* , 那么ei为A类边;

2、0＜Si＜*τ*, 那么ei为B类边;

3、Si=0, 那么ei为C类边;

根据边存在率Si的定义可知, 如果Si=*τ*, 那么边ei存在于所有的满足最大流的子图中; 如果Si=0, 表示边ei, 不存在任意一个满足最大流的子图中; 如果0＜Si＜*τ*, 表示边ei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ei, 如果边ei是A类边, 那么移除边ei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ei是A类边, 有Si=*τ*, 根据定义边ei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, …, *Cτ* }, 满足*Ci* ∩ *Cj =* ∅, *C*1 ∪ *C*2 ∪ … ∪ *Cτ* = *A*, 其中τ 为满足最大流的子图区间的个数, 因为Si=*τ* , 所以满足最大流的子图g一定有g∈A且有ei∈g, 如果当ei发生故障之后, 剩余不确定图G’的任意子图g’都不包含ei, 即ei∉g’, 则必有g∉A, 因此对于ei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Si＜*τ*, 则在A中必存在一个子图g, 使得ei∉g, 如果边ei发生故障, 不确定图可以使用子图g传递流量, 同时因为g∈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*τ*=2, 根据公式1, 可知对于边e1, 有s11=1, s12=1, 根据公式2可知, 对于边e1的边存在率S1=s11+s12=2=*τ*, 所以对于不确定图G而言, 边e1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ei, 如果边ei是B类边, 那么移除ei, 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ei, 如果边ei是B类边, 即如果有0<Si＜*τ* , 则ei移除之后, 可以找不包含边ei的子图达到最大流, 使得最大流不会下降, 但是ei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, …, *Cτ* }, 满足*Ci* ∩ *Cj =* ∅, *C*1 ∪ *C*2 ∪ … ∪ *Cτ* = *A*, 其中τ 为满足最大流的子图区间的个数, 因为0<Si＜*τ* , 则对于任意的ei∈g子图, 可以将子图的集合A分为g∈A1和g∉A2, 其中满足A1∩A2=∅, A1∪A2=A.

在ei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ei断掉之后, 集合A1中的子图不一定能够满足最大流, 但是集合A2中的流量可以满可以满足最大流, 因此保证了不确定图的连通性不变; 同时ei断掉使得A1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ei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*τ*=2, 根据公式1, 可知对于边e3, 有s31=0, s32=1, 根据公式2可知, 对于边e3的边存在率S1=s11+s12=1<*τ*, 所以对于不确定图而言e3是一个B类边, 则e3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ei, 如果ei是C类边, 那么移除边ei, 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ei, 如果ei是C类边, 即如果Si=0, 则ei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性也不会改变, 此时边ei为不确定图的非关键边.

想要证明Si=0时, ei是不确定图G的非关键边, 只需要证明当ei断掉之后, 不确定图的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, …, *Cτ* }, 满足*Ci* ∩ *Cj =* ∅, *C*1 ∪ *C*2 ∪ … ∪ *Cτ* = *A*, 其中τ 为满足最大流的子图区间的个数, 因为Si=0, 那么对于所有的子图区间每一个对应的边ei处都为标记“x”, 即任意一个子图区间可以表示为C=(c1, c2…x…cn), 其中ci表示为子图区间C的各条边对应的表示, i={1…i-1, i+…n}, 则C可以分为两个子图区间(或子图)C1=(c1, c2…0…cn)和C2=(c1, c2…1…cn), 当边ei发生故障之后, C2=C1, 因为C1中的子图都能够满足最大流, 故ei断掉之后C2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ei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e3为C类边, 因此e3移除之后, 依然可以满足最大流. 移除e3之前, 容量可靠性pc= =p(e1)p(e2)(1-p(e3))p(e4)p(e5)+ p(e1)p(e2)(e3)p(e4)p(e5)= p(e1)p(e2)p(e4)p(e5), 移除e3之后, 容量可靠性表示为pc’= p(e1)p(e2)p(e4)p(e5), 显然移除e3之后不确定图的容量可靠性不变.

### 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e5, 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e5之后的子图空间的划分.



图10 移除e5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, 同样由推论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 Ce=1 and Ce=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, …, *Cτ* }, 满足*Ci* ∩ *Cj =* ∅, *C*1 ∪ *C*2 ∪ … ∪ *Cτ* = *A*, 其中*τ* 为满足最大流的子图区间个数, 假设边ei为G的B类边, 根据B类边定义, 存在一个k∈{1, 2…*τ* }, 使得 与 , 令, 因为中的所有子图位置为0, 所以有

若要证明 为G-之后, 所有满足最大流的子图集合, 需要证明以下两点:

(1), 都有 ;

(2), 若 , 则必存在一个, 使得.

首先证明(1): 已知条件都有, 且, 所以可以推出对于任意的子图g, 如果, 都有g满足最大流, 即, 都有.

然后证明(2): 反正法. 假设, 若有, 则对于, 都有.

对于, 且有, 所以, 又因为, 所以, 这与不存在一个使得矛盾. 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, e3是一个B边, 根据以上规则, 11x11区间在e3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e3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e3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, 根据推论1, 其断掉不会影响连通性, 根据推论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照成任何的影响.

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: 根据关键程度排序好的keyEdgeSet

1. Input uncertain Graph G and its source point and sink point;
2. Get maxFlow , StateMtrix and build StateTree using algorithm SDBA
3. Get G maxflow when every single edge being cut off ALLEdgeFlowDecrease[|E|];
4. Sort keyedge by ALLEdgeFlowDecrease[|E|];
5. Classify ABC edge by StateMtrix;
6. When the flow (G-ei) equles the flow (G-ej) then
7. If ei is A class edge then use algorithm STPA\_CUT to get pd and pc;
8. If ei is B class edge then use algorithm B-CESA to get pd and pc;
9. If ei is C class edge then pd, pc is same as before ei be moved.
10. Soet KeyEdge by pd and pc;
11. Return sorted key\_edge\_set;

### 算法复杂度分析

基于状态划分树的增量算法首先也需要使用getALLEdgeFlowDecrease函数获取所有边发生故障之后能够满足的最大流, 然后使用soetKeyEdge\_by\_flow函数对计算的流量进行排序, 当前后两条边断掉之后获得的流量不相等时, 根据模型定义, 直接就可以比较出两条边的关键程度, 只有当两条边移除之后最大流一致的情况下, 会使用状态划分树的增量算法计算, 该部分的计算最大流的算法是Dinic算法, 运行Dinic算法所需的处理时间为O(|V|2|E|), 因此在处理这部分是复杂度是O(|V|2|E|2). 然后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*τ*), 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(|V|2|E|2)+O(ak|V|2|E|)+ O(b*τ*)+O(c), 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.

# 试验及分析

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V6E10、V8E14、V10E18、V12E22、V14E26共5组不同图规模的二态有向图集合(本文实验数据集合大小为5), 其中VnEm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.

**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) 不同图规模情况下BASE和ICA算法时间和内存消耗

Base\_all算法, base算法, ica算法, 实验结果和分析

**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.

[待给出]

实验分析:

# 结束语

本文构建了一种基于流量和可靠性(分布可靠性和容量可靠性)指标的不确定图关键边评估的数学模型, 使用不确定图中边移除(故障)后对流量和可靠性产生的相对损失这一角度, 对边的关键度进行综合评估. 然后设计了BASE算法作为衡量关键边的基本算法, 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, 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, 另一方面将研究对于多态不确定图的关键边评估.

References:

1. Kang H, Butler C, Yang Q, et al. A new survivability measure for military communication networks[C]//Military Communications Conference, 1998. MILCOM 98. Proceedings. , IEEE. IEEE, 1998, 1: 71-75.
2. TAN Y, WU J, DENG H. Evaluation method for node importance based on node contraction in complex networks[J]. Systems Engineering-Theory & Practice, 2006, 11(11): 79-83.
3. Newport K, Schroeder M, Whittaker G M. Techniques for evaluating the nodal survivability of large networks[C]//Military Communications Conference, 1990. MILCOM'90, Conference Record, A New Era. 1990 IEEE. IEEE, 1990: 1108-1113.
4. Nardelli E, Proietti G, Widmayer P. Finding the detour-critical edge of a shortest path between two nodes[J]. Information Processing Letters, 1998, 67(1): 51-54.
5. Nardelli E, Proietti G, Widmayer P. A faster computation of the most vital edge of a shortest path[J]. Information Processing Letters, 2001, 79(2): 81-85.
6. Callaway D S, Newman M E J, Strogatz S H, et al. Network robustness and fragility: Percolation on random graphs[J]. Physical review letters, 2000, 85(25): 5468.
7. Freeman L C. A set of measures of centrality based on betweenness[J]. Sociometry, 1977: 35-41.
8. Chen Y, Hu A Q, Hu J, et al. A method for finding the most vital node in communication networks[J]. High Technology Letters, 2004, 1(2): 573-575.
9. Xiong J, Li J, Shen D, et al. Evaluation Method for Node Importance of Information System Networks Based on Edge-betweenness[J]. Keji Daobao/ Science & Technology Review, 2013, 31(14): 53-55.
10. Cai W, Zhang B L, Lv J H. Algorithms of the most reliable maximum flow on uncertain graph[J]. Jisuanji Xuebao(Chinese Journal of Computers), 2012, 35(11): 2371-2380.
11. Klingman D, Napier A, Stutz J. NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems[J]. Management Science, 1974, 20(5): 814-821.
12. Aggarwal C C, Yu P S. A survey of uncertain data algorithms and applications[J]. Knowledge and Data Engineering, IEEE Transactions on, 2009, 21(5): 609-623.
13. Yi-Kuei Lin, Cheng-Ta Yeh. Evaluation of optimal network reliability under components-assignments subject to a transmission budget. IEEE Trans. Reliability, 2010, 59: 539-550.

附中文参考文献:

[9] 熊金石, 李建华, 沈迪, 等. 基于边介数的信息系统网络节点重要性评估方法[J]. 科技导报, 2013, 31(14): 53-55.

[10] 蔡伟, 张柏礼, 吕建华. 不确定图最可靠最大流算法研究[J]. 计算机学报, 2012, 35(11): 2371-2380.

[14] 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.

[15] 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.