**深 圳 大 学 实 验 报 告**

**课程名称：­ 算法设计与分析**

**实验项目名称： 最大流应用——论文评审问题**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 杨烜**

**报告人： 郑雨婷 学号：2021150122 班级： 高性能**

**实验时间： 2023/6/8——2023/6/15**

**实验报告提交时间： 2023/6/15**

**教务部制**

|  |  |  |
| --- | --- | --- |
| 实验目的   * + 1. 掌握最大流算法思想。     2. 学会用最大流算法求解应用问题。 | | |
| 实验内容：论文评审问题  （1）有m篇论文和n个评审，每篇论文需要安排a个评审，每个评审最多评b篇论文。请设计一个论文分配方案。  （2）要求应用最大流解决上述问题，画出m=10，n=3的流网络图并解释说明流网络图与论文评审问题的关系。  （3）编程实现所设计算法，计算a和b取不同值情况下的分配方案，如果没有可行方案则输出无解。 | | |
| 实验过程及内容：  **一、问题分析**  给定论文数m,评委数n,每篇论文需要的评委数a,每个评委评审最多评b篇论文。我们知道有且仅有在满足下面两个条件时可能有解：  ①n>=a  ②a\*m>=b\*n  因为只有评委数大于等于每篇论文需要的评委数时才有可能有解，同时还需要保证所有评委能评阅的总量大于所有论文被评审的次数，这样才可能有解。在满足了上面的两个条件后，我们可以运用最大流的思想来解决问题。  **二、构建流网络**  要运用最大流的思想解决问题，首先需要有一个流网络，下面以m=10,n=3的情况来绘制流网络图。要构建一个流网络，首先需要有源点和汇点，再将10篇论文和3个评委都抽象为点，得到了流网络中的所有节点。    图1 构建流网络  每篇论文需要安排a个评审，因此源点与论文点之间的边容量为a。每个评委最多评b篇论文，因此评委与汇点之间的边容量为b。评委和论文之间只有两种状态，一种是该评委评审了该论文，另一种是该评委没有评审该论文，因此论文和评委之间边的容量为1。将论文点与评委点两两相连，最终，构建出如下图所示的流网络图。    图2 构建流网络  流网络图与论文评审问题的关系如下：  ①在上面的流网络中求最大流，若最大流等于，说明每一篇论文都被a个评委评价过了，即为有解。若不相等(小于)，即为无解。  ②论文与评委之间的边，流量为1的代表该评委评了该论文，流量为0代表该评委没有评该论文。  因此，在上面构造的流网络中求最大流，若最大流不等于，则输出无解。反之，遍历论文与评委之间所有边，根据这些边的流量值是0或是1，得到评委是否评了这篇论文，输出论文分配的方案。  **三、求最大流的不同方法**  **相关概念：**  ①增广路径：从源点s到汇点t之间的一条路径，该路径上不存在边容量小于等于0的边。下面粗边即为一条增广路径。    图3 增广路径示意图  ②残留网络：残留网络在最大流问题中是一个非常重要的概念。它是指在原始网络上，考虑已经通过某些路径分配了一定容量后，仍然可以增加流量的那些边所组成的网络。具体来说，在一个残留网络中，对于一条有向边(u, v)，如果其原本的流量为f(u, v)，该边的容量为c(u, v)，则其剩余容量即为c(u, v) - f(u, v)；反向边(v, u)的剩余容量即为f(u, v)。例如，在图3中沿找到的增广路径压入流量为4的流时，得到残留网络图4。    图4 残留网络示意图  **3.1 Ford-Fulkerson方法**  1.方法思想：  若在残留网络中存在一条增广路径，就沿该路径压入流，流量由路径上的最小容量限制。然后再找到另一条增广路径压入流，一直到网络中不存在增广路径为止。  方法步骤：  （1）构建流网络  （2）在残留网络上寻找增广路径  （3）在流网络中的增广路径上压入流  重复（2）—（3）直到没有增广路径  2.算法实现    图5 Ford-Fulkerson方法思想示意图  基础的FF方法寻找增广路径使用的是深度优先遍历（DFS），每次DFS时将较小的容量递归传入入下一次DFS，这样直到当找到汇点时，当前的流就是这条路径上最小的容量，记为d。之后再将d压入流网络，也就是将这条路径上的正向边加上d,反向边减去d。    图6 FF方法伪代码  3.效率分析  设f\*是最大流,E是边数。在DFS过程中，每一次迭代最大流至少增大1，因此最大流的求解迭代次数至多为f\*。又因为每一轮DFS的复杂度为O(E)。基本的Ford-Fulkerson算法的时间最坏时间复杂度为：  **3.2 Edmond-Karp算法**  1.算法思想：  EK算法是FF方法的一个具体实现，整体思想和FF方法一致。知识在寻找增广路径时，它采用广度优先遍历（BFS）的方式，这样可以确保每次找到的增广路径都是长度最短的路径。  算法步骤：  （1）构建流网络  （2）在残留网络上用BFS寻找增广路径  （3）在流网络中的增广路径上压入流  重复（2）—（3）直到没有增广路径  2.算法实现  寻找增广路径使用的是深度优先遍历（BFS），这样直到当找到汇点时，就找到了一条增广路径。但与FF方法使用的DFS不同，BFS需要注意步骤（3）压入流时，无法像DFS那样递归天然地返回上一级节点，因此需要记录每个节点的前驱，在压入流update时按照前驱后继关系压入。    图7 EK算法伪代码  3.效率分析  存在如下定理：如果对具有源点s和汇点t的一个流网络G=(V,E)运行Edmonds-Karp算法，对流进行增加的全部次数为О(VE)。由于在用广度优先搜索寻找增广路径时，Ford-Fulkerson中的每次迭代都可以在О(E)时间内完成，所以Edmonds-Karp算法的全部运行时间为。  **3.3 Dinic算法**  1.算法思想：  Dinic算法的思想是分层次的在网络中寻找增广路径。先使用BFS对图进行分层，然后用DFS寻找增广路径。在Dinic算法中，我使用了多路增广进行。它与EK算法的不同之处在于：EK算法每个阶段执行完一次BFS增广后，需要重新BFS从源点开始寻找另一条增广路径。而多路增广后，只需一次DFS过程就可以实现多次增广。  算法步骤：  （1）构建流网络  （2）用BFS对图进行分层  （3）在残留网络上DFS寻找多条增广路径  （4）在流网络中的增广路径上压入流  重复（2）—（4）直到没有增广路径或者已经联通汇点。    图8 Dinic多路增广示意图  2.算法实现  首先初始化每个节点的层数都为0，用BFS对图网络进行分层，之后用DFS 寻找增广路径。注意在第一次找到汇点后不立马结束，而是继续进行DFS,如果残余流量没有用完，可以利用残余部分流量，再找出一条增广路。这样就可以在一次BFS中找出多条增广路，大大提高了算法的效率。下面图9为Dinic算法的伪代码，其中BFS与EK算法中的相似，不再给出。    图9 Dinic算法伪代码  3.效率分析  每轮BFS搭建分层图找到的增广路的数量至少为1，增广路的数量每次都减少至少一条, 整个网络中最多有V - 1条增广路,（V为顶点数量），最多V-1次，时间复杂度为O(V).分层图可以在O(E)的时间复杂度内用BFS构建。一条增广路可以在O(VE)的复杂度内构建。Dinic算法整体的时间复杂度为  **()** |
| 数据处理分析：  运用控制变量法，分别改变a,b,m,n测出三种方法的运行时间，得到图10的四幅曲线。      图10 实际运行时间  可以看到EK方法比FF慢，这是因为在论文评审问题中，图的层次是固定的，只有源点、论文、评审、汇点这四层，这样DFS效率不会很低。但EK算法使用的BFS会随着a,b,m,n的增加而宽度增加，因此效率较低。但FF方法和EK算法的效率都明显低于Dinic算法，这是因为Dinic算法中使用了多路增广，相对于单个增广路，多路增广能够更有效地利用残留网络中存在的全部潜在增广路，这种方法会大幅度减少重复搜索的情况，从而节省了计算时间。 |

|  |
| --- |
| 实验结论：  在这次实验中，我运用最大流的思想解决了论文评审问题，这个问题是实际生活中真正能够遇到的问题，说明最大流的思想可以运用在日常生活中来帮助我们解决问题。在最开始构建流网络图时，需要理解题意，将实际问题转换为数学模型是一项极为重要的能力。  在求最大流时，首先需要了解流网络中的增广路径、残留网络等概念。之后我学习到了三种方法：**Ford-Fulkerson、Edmond-Karp和Dinic。**Dinic算法的效率最高，因为多路增广可以避免重复计算，这与实验五的路径压缩思想相同，都是尽量避免做无用功。在学习了算法的思想过后，真正通过运行程序体会到三种方法的效率不同。还有这次实验又用到了DFS、BFS，这些基础知识十分重要。  通过这次实验，对于流网络的最大流有了进一步的认识，掌握了三种求解最大流的方法，收获颇丰。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。