**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： 最大流应用问题**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 杨烜**

**报告人： 谢弘烨 学号： 2020151036**

**实验时间： 2022年6月17日**

**实验报告提交时间： 2022年6月25日**

**教务部制**

|  |  |  |
| --- | --- | --- |
| **一、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。 | | |
| **二、实验原理：** | | |
| **三、实验用品：**   1. Visual Studio 2022 2. MicroSoft Office 2019 | | |
| **四、实验过程及内容：**   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。 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **五、实验现象及数据处理：**   1. 问题描述：   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。  可以用点代替论文和评审，在论文和评审之间添加有向边代表论文与评审之间的选择关系。  在此基础上可以在点集左右分别增加一个虚拟源点和汇点，即可构造一个流网络。其中源点到评审直接的连线容量代表评审最大论文数，论文到汇点的连线容量代表论文需要多少评审。原题中的可行解即代表每篇论文都被评审a次，即论文到汇点之间的容量全都流满。    至此，原题转变为求一个流网络中最大流的问题。   1. 定理证明：   残留网络中的一条从源点到汇点的路径称为增广路径，找到一条增广路径并沿这条路径增加流量的过程称为增广。  易得流网络中流量可以增大当且仅当残留网络中存在增广路径，即当流网络中不存在增广路径时达到最大流。  这就是增广路定理。基于该定理的最大流问题解法有Ford-Fulkerson方法、Edmonds-Karp算法和Dinic算法等。本报告中仅介绍前两种解法。   1. FF方法： 2. 基本思想   在残留网络中不断地寻找增广路径以增加流，直到找不出更多的增广路径为止。  Ford-Fulkerson方法之所以称为“方法”而不是“算法”，是因为它包含具有不同运行时间的几种实现。此处使用DFS进行增广路径的搜索。   1. 伪代码   Ford-Fulkerson  while there is a path in G  f\_min=min(every edge in path)  for every in edge(u, v) in path:  (u, v)-=f\_min  (v, u)+=f\_min   1. 效率分析   查阅资料可知利用DFS实现的FF方法的时间复杂度为，其中V为顶点数，E为边数。然而实际运行过程中上界往往低于查阅得到的时间复杂度。  具体编写代码实现，用材料中的例子验证：    可见算法正确   1. 方法缺陷   上述方法存在缺陷，当流网络如下图所示时：    1、2之间的路径可能会来回切换，使得仅需两次的搜索增广路径过程变成两百万次      要解决上述情况，就提出了EK算法。   1. EK算法： 2. 基本思想   大致思想与FF方法相似，仅改用BFS进行增广路径搜索。  就上述情况而言，EK算法运行情况大致如下图所示：     1. 伪代码   算法核心部分与FF方法相同，故此处不提供伪代码。   1. 效率分析   查阅资料得EK算法时间复杂度与FF方法相同，均为。然而同样的，实际运行过程中算法效率远达不到查得的时间复杂度。  具体编写代码实现，用材料中的例子验证：    可见算法正确   1. 效率比较：   当m=10，n=3时  FF方法：   |  |  |  |  | | --- | --- | --- | --- | | **a b** | **20** | **15** | **10** | | **1** | **0.18657** | **0.27894** | **0.18393** | | **2** | **0.14641** | **0.16458** | **0.12462** | | **3** | **0.13893** | **0.11854** | **0.10648** |   EK算法：   |  |  |  |  | | --- | --- | --- | --- | | **a b** | **20** | **15** | **10** | | **1** | **0.108345** | **0.108015** | **0.103245** | | **2** | **0.1957** | **0.198525** | **0.20801** | | **3** | **0.24497** | **0.2672** | **0.253095** |   数据规模增大时，由于FF方法使用DFS搜索增广路径，会进行大量重复搜索，进而导致FF方法不可用。故大规模数据仅使用EK算法。  当m，n=500时：   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **a b** | **5** | **10** | **15** | **20** | **25** | | **5** | **7113.575** | **10584.580** | **11509.750** | **12285.600** | **12638.680** | | **10** | **7150.137** | **14471.640** | **19175.150** | **20701.060** | **22094.540** | | **15** | **7176.090** | **14185.210** | **21453.740** | **25626.990** | **29543.830** | | **20** | **7158.286** | **14066.050** | **21208.030** | **27334.560** | **32516.940** | | **25** | **7207.595** | **14400.660** | **21278.870** | **27608.260** | **34346.310** |   当a=5，b=25时：   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **m n** | **100** | **200** | **300** | **400** | **500** | | **100** | **137.927** | **275.697** | **403.621** | **521.988** | **643.410** | | **200** | **486.704** | **930.375** | **1366.726** | **1818.488** | **2170.019** | | **300** | **926.586** | **1877.217** | **2855.116** | **3853.070** | **4680.389** | | **400** | **1441.751** | **3055.271** | **4723.029** | **6523.641** | **8077.952** | | **500** | **1960.704** | **4563.301** | **7194.551** | **9976.812** | **12284.167** |   可见：  m、n固定时，算法运行时间会随着b的增大而增加；而a在小于b时同样与运行时间正相关，a大于b后算法运行时间不会继续随着a的增大而增加。这是因为a、b的改变都会影响分配方案的总数目，进而使得算法运行时间增加。  a、b固定时，算法运行时间与m、n都正相关。 |
| **六、实验结论：**   1. 日常生活中一些排班问题，分配问题，以及有先后关系的选课问题都可以通过网络流的方式进行解决 2. 网络流问题解决过程中会遇到很多专有名词，了解这些名词是解决网络流问题的先决条件 3. 可以使用有多种不同思路的算法求解最大流问题。这些算法的效率也不仅相同，应该分不同情况进行解决 4. 最大流求解中的定理证明环环相扣，需要具备比较高的数学知识才能完成证明 |

|  |
| --- |
| **思考题：** |
| **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**。