黑色的仙人掌

中度可信度描述已自动生成

**计算机与信息学院**

**2024年秋季学期课程论文**

**《人工智能》**

|  |  |
| --- | --- |
| 课程类型： |  |
| 学 号： |  |
| 姓 名： |  |
| 专 业： |  |
| 授课教师： |  |

完成日期：2024年 月 日

基于网络流模型的矩阵计算问题

摘 要：网络流是一种类比水流的解题方法，与线性规划有着密不可分的关系。随着信息科学的不断发展，网络流的应用场景已经遍及运输、通信、资源分配以及各种计算机辅助设计等多种领域。在本文中，将网络流应用到了矩阵计算的问题当中，通过将矩阵模型转化成为网络流模型，通过对无源汇上下界可行流问题以及有源汇上下界可行流问题的解析，再使用Dinic算法利用层次图来求解其网络中的最大流，最后代入多个矩阵元素值的约束将所求矩阵计算出来。

**关键词：网络流；最大流；Dinic**

# 1 引言

网络流模型是一种通过网络流理论进行建模的算法模型，利用一种类比水流的解决方法来处理问题。网络流模型由两个部分组成：网络和流，其中，网络是一个带权的有向图，它拥有两个特殊的点，即源点（只有出度）和汇点（只有入度），同时它的每一条边的权值又称为容量；流则是将整个网络的边想象成一条条能通水的管道，其中的水就是流，每条边的容量则是这条管道能通过的最大水流量。而且除了源点和汇点这两个特殊点外，网络中的其他所有点的流入量都等于流出量，在这个条件的基础上，如果每条边的流量都满足它的容量限制，则可以将这些边称为可行流。通过可行流这个概念，就可以引入最大流的概念，即流入汇点的最大流量，也就是最大的可行流。而网络流模型通常都是用来求解最大流问题的。

# 2 问题描述

在本题中，给定一个数组，它的行列数，每行每列的和以及一些矩阵元素的约束，需要通过这些条件来寻找合适的可行矩阵。用 、 个点分别代表矩阵的 行和 m 列，每个顶点之间的边的流量 可以看作矩阵 的元素值 ，而矩阵的行和 就是 个行顶点的流量上下界，列和 就是 个列顶点的流量上下界。由此，可以将这个矩阵问题转化成一个有源汇上下界可行流的问题。而解决有源汇上下界可行流问题的方法通常是将这类问题转化成一个无源汇上下界可行流问题。

## 2.1 算法说明

在求解一般的网络流模型中的最大流问题时，一般都会用到以下几种算法：

### 2.1.1 Ford-Fulkerson

Ford-Fulkerson 算法简称 FF 算法，它的核心思路是寻找增广路，即一条由源点出发，最后能够抵达汇点的流量路径。如图1所示，其中 即为一条增广路[1]。在找到一条增广路后，用这条路上的每一条边的容量减去其中最小边的容量，此时 的容量由3变成了1， 的容量由2变成了0，这时的容量称为残余容量。后面依次类推，找出这个模型中的所有增广路。

图示

描述已自动生成

图1 普通的网络流模型

由于一个网络流中会存在多条增广路，这样在找到一条增广路后，可能不是整个网络的最大流，这样就与需求相悖。所以在这里引入反向边。在寻找到一条增广路并扣除正向边的容量后，在原路的基础上用反向边建立另外一条路，同时在每一条反向边上加入等量的容量。若当前的增广路并非最大流，可以通过反向边进行一次“撤回”，通过这种机制就可以保证，当网络流中的增广路找不到后，从源点流向汇点的流量就是最大流。

### 2.1.2 Edmonds-Karp

EK 算法是基于上面的 FF 算法的最短增广路算法，在原有的算法基础上，通过广度优先搜索算法（BFS）来寻找增广路。将先构建一个残量网络流模型，再将所有残量为正的边看作可行边，然后在这个模型用运行一遍 BFS 就可以找到最短的增广路[2]。

### 2.1.3 Dinic

由上述的 EK 算法可知，该算法要对增广路进行多次寻找，这样会导致算法的时间复杂度很高，效率会很低。因为就提出了 Dinic 算法。虽然都是寻找最短增广路，但与 EK 算法不一样的是，Dinic 算法先通过广度优先算法 BFS 对网络流模型进行分层，得到一共分层图，再通过深度优先搜索算法 DFS 找到当前所有的增广路后，再对残量网络模型进行分层，再用DFS 进行寻找，直到无法分层，也就是再找不到一条增广路了。

## 2.2 无源汇上下界可行流

在这个问题中，可以从名称中得知它的特点：在一个网络流中没有源点和汇点，同时每一条边相对于普通的网络流模型来说，多了一个下界来约束其流量，而且是一个求可行流问题。

在这个特殊的网络流模型中，解题思路就是将它转化成一个普通的网络流模型，在对这个普通的网络流模型进行求解[3]。设每条边的流量上下界分别为： 和 ，所以由模型的流量约束可表示为 ，而在普通的网络流模型中，其流量约束表示为 。因此，将上下界可行流约束中的上界和下届以及流量同时减去下界值，就可以得到需要的形式：，再将 看作是另一个网络流模型中的流量， 看作是它的容量，就可以得到一个新的普通网络流模型。

因为在原来的网络中是满足流量守恒的，对于网络中的某一中间点 ，它的出度等于入度，即 ，而在新的网络中，每条边的流量发生了变化，所以新网络不一定满足流量守恒，因此需要对新的 和 两者的大小进行判断。若 ，则需要建立一条由源点到 的边，其容量为 ；若，则需要建立一条由 到汇点的边，其容量为 。由此可知，在新网络中，当源点到 的边必须是满流的才能达到流量守恒。所以若从源点流出的边均是满流的，则原网络中就存在可行流，进而将求解原网络中的可行流问题转化成了求新网络中的最大流问题。

## 2.3 有源汇上下界可行流

这个问题与上述的无源汇上下界可行流类似，是在无源汇的问题模型的基础上增加了源点和汇点。所以解决这个问题的思路和上述问题的思路也类似，通过将有源汇上下界可行流问题的模型转化为一个普通的网络流模型问题。

图示

描述已自动生成

图2 有源汇模型的更改模型

首先需要使网络满足流量守恒，因此在网络中添加一条从汇点到源点的边，其容量为无穷大，如此，无论有多少流量从源点流出，再流向汇点，都可以再从汇点返回至源点[4]。此时为了使源点和汇点满流，就需要再额外添加一个附加源点和一个附加汇点。如图2所示，因为 的容量为3，但 的容量为1，使得 最多只能流入的量为1，而且使得流入 的流量也为1，都使得源点和汇点均无法满流，这样就无法判断原网络中是否存在可行流，因此在原图的基础上，增加附加源点 和附加汇点 ，再添加两条流入 的边 ，容量为1 和 ，容量为2；以及一条流出 ss 的边 ，容量为3，这样就使得从源点出发流入汇点的边都是满流的，从而可以判断原网络中存在可行流，同样也将此问题转化成最大流问题。

# 3 问题解析

根据第二节问题描述的说明，可以先根据给定的行列数确定行节点个数 以及列节点个数 ；再用一个节点 连接所有的行节点，边的容量分别为每行的和，将 当作源点；用另一个节点 连接所有的列节点，边的容量分别为每列的和，将 当作汇点；之后再将所有的行节点和列节点相连接，每条边的容量为无穷大。这样就构建好了一个求最大流问题的模型，再求 到 的最大流。如果从 流出的每一条边都是满流的，即该问题有解，每一个行节点 连接列节点 的边的流量就是矩阵节点 的元素值。

示例如图3所示，其中 ， 以及行和分别为 ，，列和分别为 ，，。

墙上的钟表

低可信度描述已自动生成

图3 矩阵问题转化为求最大流问题示例

在将矩阵问题转化成求解有源汇最大流问题时，选择效率比较高的Dinic算法来求解。首先需要对 Dinic 算法进行实现。根据第二小节中的算法说明可以知道，在Dinic算法中，需要使用广度优先搜索对模型进行分层，在使用深度优先搜索进行寻找。所有要实现该算法就需要先对BFS和DFS进行实现。

文本

描述已自动生成

图4 广度优先搜索

广度优先搜索算法的思想是从一个图中的某一点 开始，先访问 ，再访问所有与 相邻的节点 ，，…，，在依次访问 ，，…， 的所有相邻节点，直到找到目标节点或者访问完所有节点位置[5]。由此可见，BFS的搜索次序是沿层次向外横向展开的，因此可以通过在网络中使用BFS算法对网络进行分层处理，也就是预处理从源点出发到每一个节点的距离，在分层之后，只要每次依次选择层数高的方向，就能够防止往回走或者绕圈子。在Dinic算法中的BFS实现如图4所示。

文本

描述已自动生成

图5 深度优先搜索

将网络转化为分层图后，就需要用到深度优先搜索算法根据所分的层次一层一层进行搜索，寻找增广路。DFS的思想是从图中的某一点 开始访问，一直沿着从 开始的一条路一直走下去，如果走到底了而未得到寻找到目标节点，就返回到上一个节点，再沿着这个节点一直往下走，访问其他未访问过的节点，直到图中所有节点被访问完或者找到目标节点为止[6]。Dinic算法中的DFS实现如图5所示。

基于以上两种搜索算法，根据Dinic算法的思路，就可以对此算法实现，Dinic算法的主要过程如图6所示。

文本

描述已自动生成

图6 Dinic算法主过程

回到这个问题本身，在构建如图3所示的模型之后，下一步就是将其转化为一个求无源汇上下界可行流问题。

因为图3中网络中的每一条边都没有下界，所以在添加一条从汇点流向源点的边 后，再给所有从 流出的与行节点相连的边添加下界，其大小等于其上界，也就是其容量；再给所有流向 的与列节点相连的边添加下界，其大小也等于上界。如此，就使得该网络变成了一个有源汇、存在上下界的网络，根据第二节中提到的有源汇上下界可行流问题的解题思路进行求解。

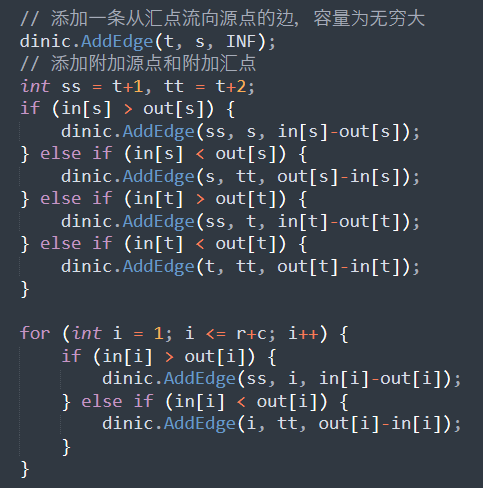


图7 流量守恒过程

在这个问题中，有一个非常重要的条件就是流量守恒。因此在网络添加一条从汇点流向源点的边，其容量为无限大，进一步将该网络转化为一个无源汇，存在上下界的网络，因此也将问题转化成了一个求无源汇上下界可行流问题。为了使得这个问题有解，就需要满足从源点流出的边均是满流的，所以还要在网络中加入一个附加的汇点让此条件成立，还需要加入一个附加源点让整个网络达到流量守恒。其过程如图7所示。

因为本问题还针对矩阵中的某一具体的元素给出了相应的约束，所有还需要对其约束进行判断，其约束的输入格式为 。具体过程如图8所示，首先判断该节点所在的边是否连接源点或汇点，再对运算符号进行判断，通过约束值与上下界的值进行比较来确定该节点的元素值。图中只展示了 这一种情况，还有两种情况 以及 ，其思路和这种情况类似，只是运算符判断中多了一层循环计算。

文本

描述已自动生成

图8 对约束条件的判断

最后，通过Dinic算法求出附加源点 到附加汇点 的最大流，通过这个解，再结合给出的元素值约束，最终求出一个可行矩阵。

# 4 总结

在完成这道题的过程中，让我学习到了很多相关的知识，都是从前所不了解的。虽然课堂上老师关于网络流问题已经讲得非常详细了，但是因为课后并没有进行相关的练习，所以对网络流的一些概念等并未了解得十分深刻。而通过这次的作业，我对网络流以及其应用等进行了深刻的了解和学习，同时通过对其算法的实现过程，又加深了从前学过的算法知识并加强了我的代码编写能力。

通过算法设计与分析这门课程的学习，扩大了我对算法理论的掌握范围以及它们的相关应用场景。从事计算机相关的学习，算法是一门十分重要的必修课。在很多领域中，算法都起到了非常关键的作用，一个好的算法往往能够改变一个程序的细节，让其性能发生天翻地覆的变化。而算法也是一门无穷尽的学科，在计算机学科发展的历史长河中，依旧存在许许多多还未解决的算法难题，因此在这门课程的学习过程中，加强了我对算法学习的兴趣，希望在未来的学习过程中，我亦能够通过在对算法的无穷尽的学习研究中获益匪浅。

**参考文献**

[1]田丰, 张运清. 图与网络流理论[M]. 科学出版社, 2015.

[2]高随祥. 图论与网络流理论 : Graph theory and network flow theory[M]. 高等教育出版社, 2009.

[3]张宪超, 陈国良, 万颖瑜. 网络最大流问题研究进展[J]. 计算机研究与发展, 2003, 40(9):12.

[4]林诒勋. 线性规划与网络流[M]. 河南大学出版社, 1996.

[5]温静. 通过广度优先搜索有向图重组规则次序[J]. 湖北大学学报：自然科学版, 2004, 26(4):3.

[6]龚建华. 深度优先搜索算法及其改进[J]. 现代电子技术, 2007, 30(22):3.

[7]凌永发, 徐宗本. 一种求解网络最大流问题的算法[J]. 计算机科学, 2006, 33(6):3.

[8]党耀国, 刘思峰, 方志耕. 网络最大流的割集矩阵算法[J]. 系统工程理论与实践, 2003, 23(9):125-128.

[9]Shimon, Even, R, et al. Network Flow and Testing Graph Connectivity[J]. SIAM Journal on Computing, 1975, 4(4):507–518.

|  |  |  |  |
| --- | --- | --- | --- |
| **项目** | **分值** | **评价依据** | **评分** |
| 论文格式及书写 | 10 | 论文格式是否严格按照规定执行，论文中的术语、格式、图表、数据、公式、引用、标注及参考文献均符合规范要求，字体撰写是否工整认真。 |  |
| 论文结构与文字  表达 | 40 | 论文结构是否严谨，逻辑性如何，论述层次是否清晰，语言是否准确；内容主题是否鲜明，是否符合课程教学内容。 |  |
| 论文内容 | 50 | 论文内容是否紧扣结课论文主题，篇幅是否达到结课论文要求，参考文献充分、正文引用恰当，是否坚持课程论文原创，有无抄袭剽窃现象。所得结论是否属实，所列举实例是否完整，代表性如何。论文是否有创新性成果或独立见解。 |  |
| 总分 | | |  |
| 评语：  论文格式严格按照规定执行，结构很严谨，逻辑性很强，论述很清晰。    　　　　　　　　　　　　教师签名： 日期： | | | |