

# 云南大学数学与统计学院

## 《算法图论实验》上机实践报告

课程名称：算法图论实验	年级：2015 级	上机实践成绩：
指导教师：李建平	姓名：刘鹏	专业：
上机实践名称：二部图的最小匹配问题	学号：20151910042	上机实践日期：2018-12-18
上机实践编号：8	组号：	

### 一、实验目的

1. 了解最小匹配问题及其解决算法；
2. 了解最小匹配算法的实际应用模型。

### 二、实验内容

1. 用网络流算法实现求二部图的最小匹配问题，用伪代码将此算法表示出来；
2. 用 C 语言实现上述算法。

### 三、实验平台

Windows 10 Pro 1803；

MacOS Mojave。

### 四、算法设计

**定义 1 匹配** 给定  $G = (V, E)$ ，若  $M \subseteq E$ ，且  $M$  中任何两条边都不相邻，则称  $M$  是  $G$  的**边无关集**（Edge-independent Set），也称为**匹配**（Matching）。我们称在  $M$  中的边为  $M$ -边，不在  $M$  中的边称为非  $M$ -边。如果点  $v \in V$ ，且  $\Phi(v)$  中存在  $M$ -边，则称  $v$  为  $M$ -**饱和点**（M-saturated Vertex）；反之，如果  $\Phi(v)$  中全是非  $M$ -边，则称  $v$  为  $M$ -**非饱和点**（M-unsaturated Vertex）。

**定义 2 完美匹配** 给定  $G = (V, E)$ ， $M$  是  $G$  的一个匹配，而且  $G$  中没有  $M$ -非饱和点，则称  $M$  是  $G$  的**完美边无关集**（Perfect Edge-independent Set），也称  $M$  是  $G$  的**完美匹配**（Perfect Matching）。如果对于边无关集  $M$ ，不存在另一个边无关集  $M^*$ ，使得  $|M^*| > |M|$ ，那么称边无关集  $M$  是**最大的**（Maximum）；**完美边无关集一定是最大边无关集，但是最大边无关集不一定是完美边无关集。**

最大边无关集不一定是完美匹配，换言之，在某个合适这种情况的图中，存在一个和完美匹配同势的另一个不同的匹配  $M$ ，亦即在该匹配中存在  $M$ -非饱和点。

#### 4.1 问题描述

**问题 1 描述：**给定一个二部图（不一定是正则二部图），通过算法计算，如果这个图有完美匹配，则输出一个完美匹配，否则输出该图不含完美匹配。

**问题 2 描述：**给定一个边赋权二部图，且所有权均非负，输出该图的具有最小重量的最大边无关集。

#### 4.2 问题 1 与问题 2 的审视

由于最大边无关集不一定是完美边无关集，所以无法通过网络最大流一般的给出算法，但是第二个问题

可以使用网络流算法来解决。除此之外，比较一般的算法是匈牙利人给出的匈牙利算法。由 Berge 定理可知，边无关集  $M$  是图  $G$  最大边无关集，当且仅当  $G$  中不含  $M$ -增广链。

### 4.3 求二部图 $B$ 的最大边无关集算法

首先根据 Berge 定理，给出求二部图的增广路的算法：

**Algorithm** 求二部图的增广链，记此算法为  $R$

**Input** 二部图  $B = (S, T; E)$ ，图  $B$  的一个匹配  $M$ ，如果是零匹配也允许

**Output** 图  $B$  的一个增广链  $C$ ，记  $C = R(B, M)$

**Begin**

**Step 1** // 得到  $S$  中的非饱和点集合

$X = \{v \in S \mid v \text{ is } M\text{-unsaturated vertex}\}$

**Step 2** // 初始化染色

**for each vertex**  $v \in X$ :

**for each vertex**  $u \in (B.V - \{v\})$

$u.\text{color} = \text{White}$

$u.d = \infty$

$u.\pi = \text{NIL}$

**Step 3** // 起点染色，初始化队列

$v.\text{color} = \text{Gray}$

$v.d = 0$

$v.\pi = \text{NIL}$

$Q = \emptyset$

ENQUEUE( $Q, v$ )

**Step 4** **while**  $Q \neq \emptyset$ :

$u = \text{DEQUEUE}(Q)$

**if**  $u \in S$ :

$J = \{x \in B.\text{ADJ}[u] \mid (u, x) \text{ is not } M_{\text{edge}} \text{ and } x \text{ is White}\}$

**if**  $J = \emptyset$ :

**continue**

**else:**

**for each**  $x \in J$ :

$x.\text{color} = \text{Gray}$

$x.d = u.d + 1$

$x.\pi = u$

                ENQUEUE( $Q, x$ )

**else if**  $u \in T$ :

```

if  $u$  is  $M$ -unsaturated:
    return PATH( $u \rightarrow v$ )
    goto End
else:
     $J = \{x \in B. \text{ADJ}[u] \mid (u, x) \text{ is } M_{\text{edge}} \text{ and } x \text{ is White}\}$ 
    if  $J = \emptyset$ :
        continue
    for each  $x \in J$ :
         $x.\text{color} = \text{Gray}$ 
         $x.\text{d} = u.\text{d} + 1$ 
         $x.\pi = u$ 
        ENQUEUE( $Q, x$ )
     $x.\text{color} = \text{Black}$ 

```

**End**

这个算法在 Berge 定理的基础上，对于反圈法进行了修改，使得反圈的修改变得有了条件，然后就可以找寻增广路。如果找到了一个增广路，就立刻停止算法，然后输出，输出之后就可以计算  $M^* = M \oplus E(\text{PATH}(u \rightarrow v))$  就可以得到一个更大的匹配；然后再计算  $R(G, M^*) \dots$  直到再也找不出增广路。这是个渐进式最优化算法。这个算法与网络最大流算法的核心思想是完全一样的，只不过后者需要构造一个剩余网络。

该算法借鉴了反圈法，但是并不能把反圈法当作子算法。

## 五、 程序代码

详见电子版。

## 六、 参考文献

- [1] 林锐. 高质量 C++/C 编程指南 [M]. 1.0 ed., 2001.
- [2] Renfei Song, 二分图的最大匹配、完美匹配和匈牙利算法, Aug. 1, 2013 / 算法, <http://www.renfei.org/blog/bipartite-matching.html>