091M4041H - Assignment 5

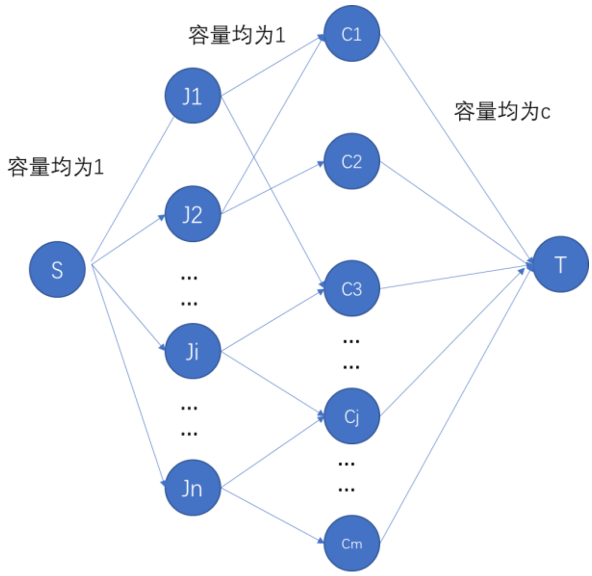
胡鹏飞 计控学院 201828007329009

1 Load balance

You have some different computers and jobs. For each job, it can only be done on one of two specified computers. The load of a computer is the number of jobs which have been done on the computer. Give the number of jobs and two computer ID for each job. You task is to minimize the max load.

1.1 Describe in natural language

假设有 n 个 job 以及 m 台 computer，根据题意，每个 job 与两台 computer 之间构建连边，容量为 1；有一个源点 s 与每个 job 都构建一个连边，容量也为 1；所有 computer 都汇于一点，每条连边的容量为 c，c 为 computer 中单独一台的最大容量。如下图 G 所示



其中 J1、J2……Jn 表示 job，C1、C2……Cm 表示 computer，S 为源点，T 为汇点，c 为 max load。

由题意可知 c 的范围为[1,n]，所以需要进行 log n 次最大流算法。

1.2 Pseudo-code

```
MinMaxload(G)
    b=1,e=n
    while b<e
        c=(b+e)/2
        if exist a flow f',satisfy max_flow(s,t)==n
            e=c
        else
            b=c+1
        end if
    end while
return b
```

1.3 Prove the correctness

首先，对于节点 J，流入的值为 1，根据网络流特点，流出的值也为 1，所以每个 job 最

后只会在一台 computer 上工作，所以采用最大流算法，如果汇点 T 可以输入 n，那么说明此时 max load，也就是 c 偏大，继续缩小，直到汇点 T 的输入不能达到 n，说明此时有一部分由于超过 c 而无法流到 T，也就是找到了最小的 c，即 max load，算法无误。

1.4 Analyse the complexity

共需要 log n 次网络流，节点个数为 n+m+2，连边条数为 n+2n+m=3n+m，所以每次网络流算法时间复杂度为 O[(n+m+2)(3n+m)]所以时间复杂度为：

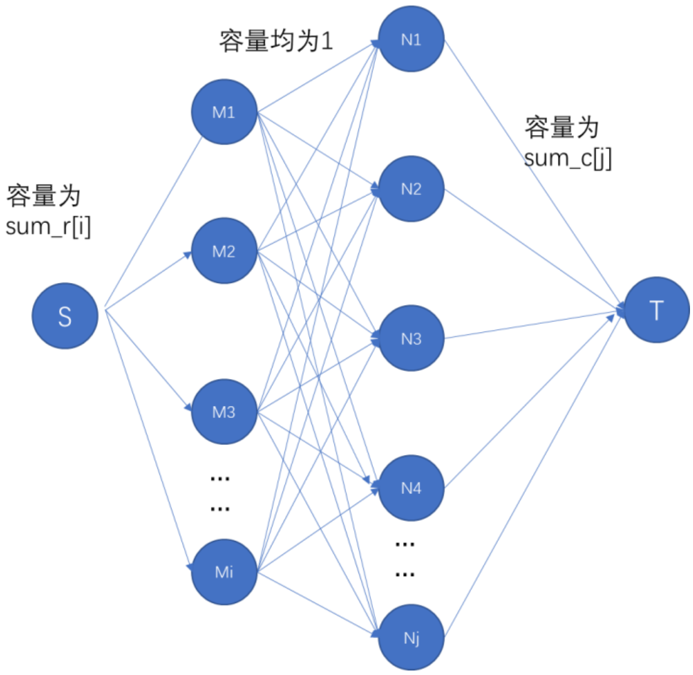
$$T(\mathrm{n}) = O[\log(\mathrm{n}) * (n+m+2)^{\wedge} 2 * (3n+m)]$$

2 Matrix

For a matrix filled with 0 and 1, you know the sum of every row and column. You are asked to give such a matrix which satisfys the conditions.

2.1 Describe in natural language

假设矩阵有 m 行 n 列，每一行构造一个节点，有 m 个，每一列也构造一个节点，有 n 个，这 m 个节点与 n 个节点都互连，每条连边容量为 1；有一个源点 s 与所有行节点都构建一个连边，容量为改行对应的和；所有列节点都汇于一点，每条连边的容量为该列的和。如下图 G 所示



跑最大流算法，如果最大流的结果等于各行的总和（各列的总和），则问题有解。

2.2 Pseudo-code

```
Matrix(m,n,sum_r[m],sum_c[n])
    constructing a graph G
    running max_flow Algorithm in G
    if max_flow(s,t) == sum of sum_r or sum of sum_c
        return the final flow graph
    else
        return False
   end if
```

2.3 Prove the correctness

源点到 Mi 的容量为第 i 行总和，保证了各行总流量不会超过 sum_r[i], Nj 到汇点的容量为第 j 列总和，保证了各列总流量 sum_c[j]，Mi 到 Nj 的容量全部设为 1，意味着只能为 0 或 1，与矩阵每个位置只能填充 0 或 1 对应。因此，若最后最大流等于所有行或所有列总和，那么矩阵有解，返回求得的 flow graph 即可, 算法无误。

2.4 Analyse the complexity

节点个数为 n+m+2，连边条数为 mn+m+n，所以所以时间复杂度为

$$T(\mathrm{n}) = O[(n+m+2)^\wedge 2*(mn+m+n)]$$

4 Network Cost

For a network, there is one source and one sink. Every edge is directed and has two value c and a. c means the maximum flow of the edge. a is a coefficient number which means that if the flow of the edge is x, the cost is ax2.

Design an algorithm to get the Minimum Cost Maximum Flow.

4.1 Describe in natural language

如果每条边的费用固定，那么这个问题就是标准的最小割最大流问题，现在又多了一个系数 a 用来计算费用，且费用与此条边的流量有关，cost=ax^2，但是可以将这个问题转化，可以将此处容量为 n,费用为 an^2 的一条边转换为 n 条边，每条边的容量为 1，费用分别为 a,3a,5a……(2n-1)a，再跑标准算法即可。

4.2 Pseudo-code

```
MinCost_MaxFlow(G):
    constructing a graph G' From G
    running Min_Cost_Max_Flow Algorithm in G’
 return Mincost
```

4.3 Prove the correctness

证明图 G 与图 G’ 等价即可，图 G 中一条容量为 n，费用为 an^2 的边转换为图 G’ 中容量为 1,费用分别为 a,3a,5a……(2n-1)a 的 n 条边,总容量仍为 1*n=n,费用仍为 a+3a+5a+……+(2n-1)a= an^2,所以两图等价，算法无误。

4.4 Analyse the complexity

寻找最短路径算法复杂度为 O(VlogV),所以总时间复杂度为:

$$T(\mathrm{n}) = O(\mathrm{V}^3\mathrm{ElogV})$$