

Yongzuan Wu

wu68

cs573

HW 5

Problem 2

reference: <http://www.cs.princeton.edu/wayne/cs423/lectures/max-flow-applications-4up.pdf>

We reduce the problem to a maximum flow problem with edge demand. First we calculate the sum of each original rows and columns, call them $rowsum[1..m]$ and $colsum[1..n]$. We use a vertex to represent each row and column, and edge between row i and column j represent the number $A[i][j]$. Formally,

let $G = (R \cup C \cup \{s, t\}, E, c, d)$ be a flow network with edge demand, where $R = \{r_1, r_2, \dots, r_m\}$, $C = \{c_1, c_2, \dots, c_n\}$,

$$c(u \rightarrow v) = \begin{cases} rowsum[i] & \text{if } u = s \text{ and } v = r_i \\ colsum[i] & \text{if } u = c_i \text{ and } v = t \\ \lceil A[i][j] \rceil & \text{if } u = r_i \text{ and } v = c_j \\ 0 & \text{otherwise} \end{cases}$$
$$d(u \rightarrow v) = \begin{cases} \lfloor A[i][j] \rfloor & \text{if } u = r_i \text{ and } v = c_j \\ 0 & \text{otherwise} \end{cases}$$

A integer value of flow in this network represents an appropriate rounding of elements of the matrix. And the maximum flow of an integer capacity network is always integer. Thus the algorithm follows,

```
MatrixRounding(A[1..m][1..n])
  construct flow network  $G$  as described above
  compute maximum flow  $F$  in  $G$ 
  if ( $F$  saturates all the edges going out of  $s$  and all the edges going into
 $t$ )
     $A'[i][j] = f(r_i, c_j)$ 
    return  $A'$ 
  else return ('No such rounding exist')
```

Analysis: Computing the maximum flow by Dinitz algorithm with dynamic trees costs $O(VE \log(V))$ time. By construction, $|V| = O(m + n)$, $|E| = O((m + n)^2)$. The cost to construct the network and convert the edge demand problem to a normal problem is relatively small, thus absorbed in the big O notation. Thus the running time is $O((m + n)^3 \log(m + n))$.