Implementing NRR

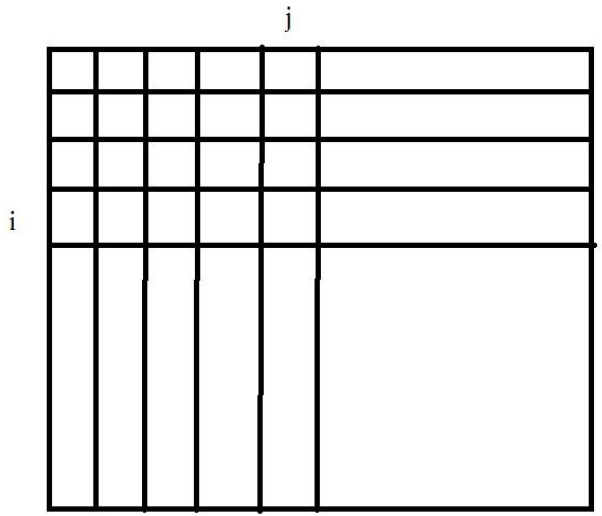
An Algorithmic Approach

Outline

1. Adjacency Matrix
2. Procedure / Algorithm - Backward Search
3. Procedure / Algorithm - Forward Search
4. Remarks
5. Future Work
6. Allied Work

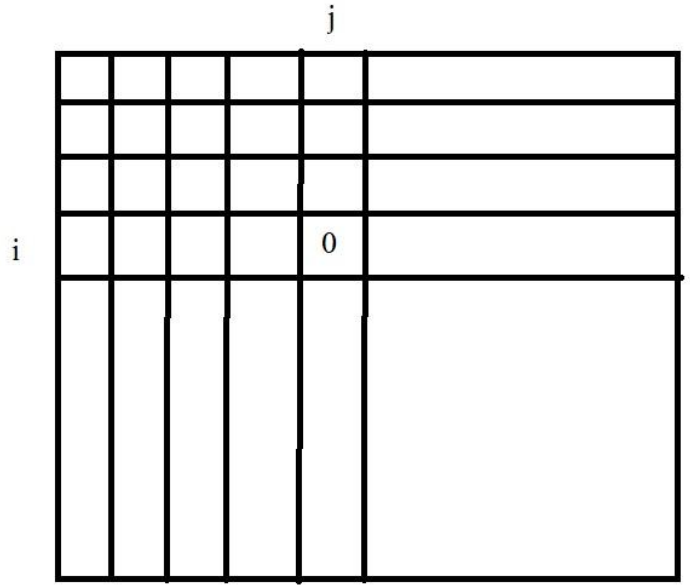
Adjacency Matrix

- An adjacency matrix for a graphical network contains a `1’ in position (i,j) if node i is directly linked with node j via an edge (direct connection)
- We want to use the adjacency matrix to study indirect connections.



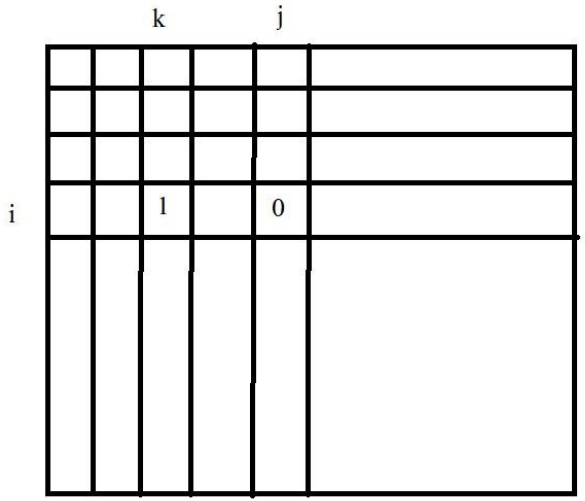
Procedure / Algorithm

- If there is a `0’ in (i,j) it means there is no direct connection between nodes i and j



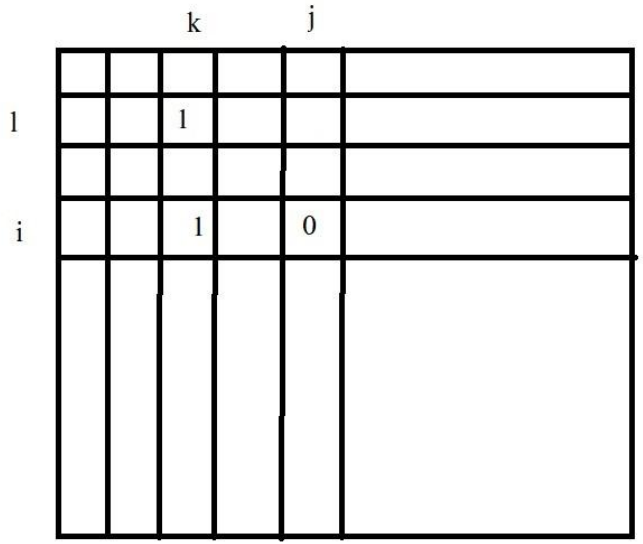
Procedure / Algorithm 2 - Backward Search

- Step 1: Consider row i, columns 1 through j - 1
- Step 2: If there exists a 1 there, go to Step 3, else go to Step 4’
- Step 3: Note down the column number of the `1.’ Say it is column k, where 1 <= k <= j.



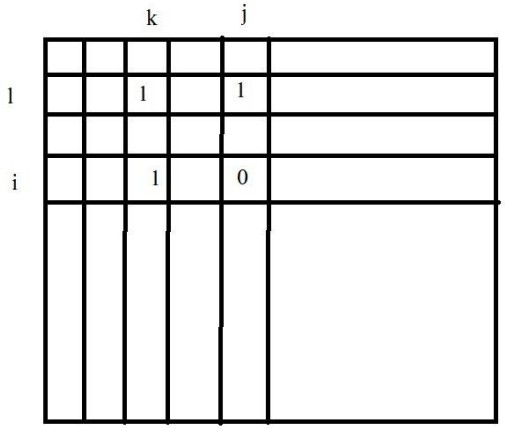
Procedure / Algorithm 3

- Step 4: In column k, consider rows i-1 through 1 in descending order (bigger value to smaller value). If there is a 1 encountered there, note its row number. Say it is l.



Procedure / Algorithm 4

- Step 5: From position (l,k), scan row l, columns k through j. If position (l,j) is a 1, then stop, since there is an indirect path between i and j. The path is (i,k) → (k,l) → (l,j). It is a four node path, with three edges.
- Step 6: If no 1 is encountered in Step 4, there is no indirect connection yet found.
- Step 7: If more than one `1' is encountered in Step 4 say on rows l1, l2, … and Step 5 using the l1 1 was unsuccessful, repeat Step 5 using the second such `1', namely the one in row l2 column k.



Procedure / Algorithm 5 - Forward Search

- Step 4’: Consider row i, columns j+1, through n. If there exists a `1’ therein, go to Step 5’
- Step 5’: Note down its column number. Say it is k’ where j+1 <= k’ <= n.
- Step 6’: In column k’ consider rows i+1 through n. If a `1’ is encountered, note its row number. Say it is l’.
- Step 7’: From position (l’, k’) scan row l’. If column j has a `1’ then we are done.

Remarks

1. Primed steps represent a `Foward' search
2. Unprimed steps represent a `Backward' search
3. The adjacency matrix A is sufficient to determine graph `connectivity' which includes indirect connections.
4. This algorithm will determine at least one ``3-edged" indirect path between nodes i and j, if it exists.
5. Versions of this algorithm can be created in future to determine at least one ``w-edged" indirect path between nodes i and j if it exists.

Future Work

- Bimodal version
- NRR full implementation:
  - Determine every connected pair (i,j) (direct or indirect)
  - Delete all edges
  - Randomly assign routes for indirectly connected pairs

Allied Work

1. Capacity analysis
   a. Suppose 1s are replaced by link capacities. Can we use this algorithm to compute max throughput between a pair (i,j)?
   b. Compare with max-flow/min-cut theorem
2. Performance analysis
   a. Instantiate a random graph
   b. Serve it as input to NRR
   c. See time taken to perform NRR
   d. Can random matrix theory plus Lovasz's paper on `Random walks on graphs' be used to get an analytical result on speed and performance?