Set 2 Algoritmi

Cracco Andrea VR432685
Alberto Purrone VR433409
Litterini Simone VR439416

13 dicembre 2018

1 Exercise 1

Provide a linear time algorithm that takes in input a weighted directed graph $G = (V, E)$, with costs on theedges $c(e)$ for $e \in E$ (cost may be negative), a node $t \in V$ and, for each $v \in V$ a value $d(v)$,and decides whether for each $v \in V$ it is true that $d(v)$ is the cost of the path of minimum cost among all the paths from from v to t.

1.1 First Observation

We change the cost of each edge $e(v, w)$ according to the formula

$$c_{new}(e) = c(e) - d(v) + d(w)$$

- if $c_{new}(e) < 0$ the minimum cost didn't take in account the edge because $d(w) - d(v)$ is greather than the cost of the current edge so the cost are not correct;
- if $c_{new}(e) = 0$ the minimum path from v to t passes through this edge;
- if $c_{new}(e) > 0$ the edge is not part of the minimum path from v to t;

1.2 Second Observation

If there exist a path from v to t of cost c and we reverse all the edges, there should be a path from t to v with cost c;

1.3 Algorithm

1. Suppose that if t is not reachable from a vertex v, $d(v) = +\infty$;
2. For each edge $e(v, w)$, change its cost to $c_{new}(e)$;
3. Reverse the direction of all edges;

4. Starting from t, do a modified DFS in the following way:
5. For each edge $e(v, w)$ that is outgoing from the current vertex v:
   - If $c_{new}(e) = 0$: visit the node w
   - If $c_{new}(e) < 0$: return false
   - If $c_{new}(e) > 0$: ignore the edge
6. If there exist a node v not visited by the algorithm such that $d(v) < +\infty$: return false;
7. Return true;

2 Exercise 2

You are given a weighted directed graph $G = (V, E)$, with costs on the edges $c(e) fore \in E$ (cost may be negative), a node $t \in V$ and, for each $v \in V$ a value d(v) which is the cost of a path of minimum cost from v to t. Provide an $O(|E|log|V|)$ algorithm that given G, $d()$ and a node t 6 = t 0 computes, for each $v \in V$ the minimum cost of a path from v to t 0.

2.1

3 Exercise 3

Show that if we apply the algorithm seen in class to find the min-cost circulation for the network shown in the figure, there exist a sequence of choices of the cycle along which we improve that require $2 * 10^6$ iteration to solve the problem.

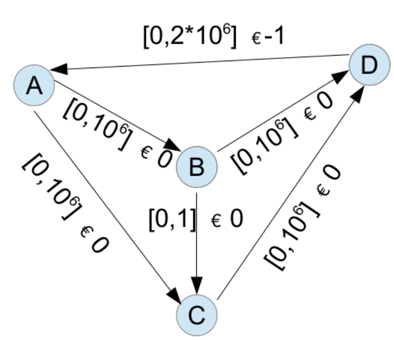


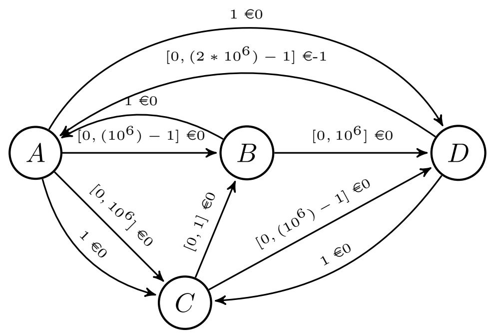
Figura 1: Flow network(N) of the problem

3.1 Algorithm

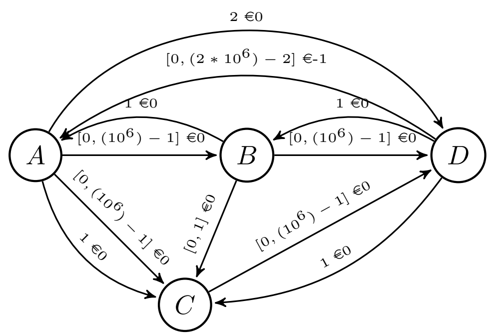
The steps of the algorithm saw in the class are:

1. Find a Feasible circulation Fc for N
2. While $R(Fc)$ has no negative cycle C, then $Fc = Sw(Fc, C)$

The fist step of the algorithm saw in the class is to produce the residual graph



the graph above is the result of the first iteration of the worst choice that the algoithm can do.



the second graph is the second iteration and if we continue to take the path from A to D passing each time from the edge between B and C (the path ABCD and ACBD) the result is a repeat of this two iterations above and using this choice we have that the algorithm takes $2 * 10^6$ iterations

4 Exercise 4

Assume that the following matrix has to be tested for the C1P. Using the algorithm seen in class provide

- a The overlap graph
- b The containment graph

If the matrix has the C1P, provide 3 distict permutations of the columns that leave the 1s consecutive in each row (if there are fewer than 3 possible such permutations, provide them all). If the matrix does not have the C1P then apply the gap minimization algorithm considering only the last 5 columns.

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4.1 overlap graph

for the overlap graph we have to compute Si that is the set of the columns that the value is 1; For the matrix A:

- r1 -> S1={1,2,3,5,6,8,9}
- r2 -> S2={1,2,3,4,5,6,9}
- r3 -> S3={2,3,5}
- r4 -> S4={2,6,9}
- r5 -> S5={1,2,3,5,6,7,8}
- r6 -> S6={2}
- r7 -> S7={5}

then:

- r1 overlap with r2
- r1 overlap with r5
- r2 overlap with r5
- r3 overlap with r4
- r4 overlap with r5

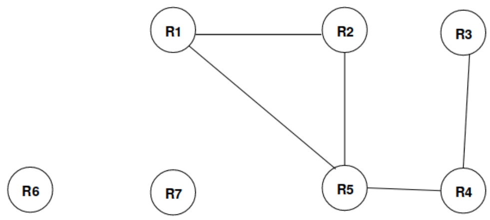


Figura 2: overlap graph

4.2 containment graph

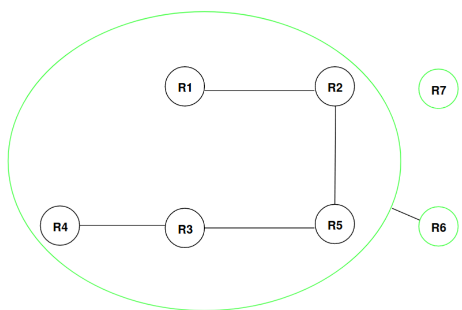


Figura 3: containment graph

4.3 Permutation of A

The only 2 permutations of the matrix A(c1,c2,c3,c4,c5,c6,c7,c8) in C1P are the following:

- c4 c9 c6 c2 c3 c5 c1 c8 c7

$$perm_1(A) = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

- c4 c9 c6 c2 c5 c3 c1 c8 c7

$$perm_2(A) = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5 Exercise 5