Bioinformatics III

Fourth Assignment

Thibault Schowing (2571837)
Wiebke Schmitt (2543675)

May 8, 2018

Exercise 4.1: Dijkstra's algorithm for finding shortest paths

(a) *Draw a directed or undirected graph with at least one negative edge weight for which Dijkstra's algorithm does not find the shortest path from some node s to another node t. Use your example to explain why Dijkstra's algorithm only works on graphs with non–negative edge weights.*

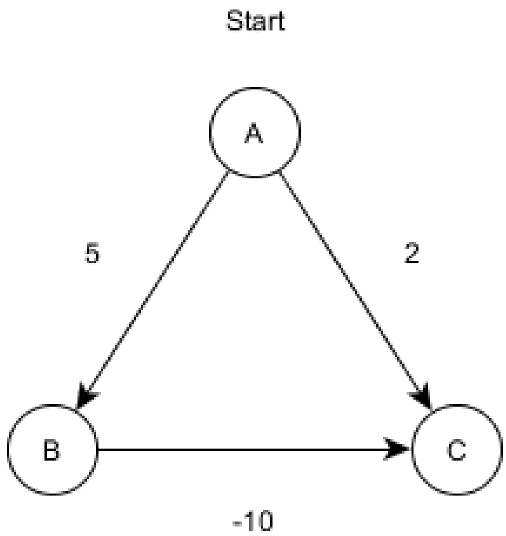

Figure 1: Example directed graph

A-B-C is the shortest path.

We have $V = A, B, C$, $E = (A, C, 2), (A, B, 5), (B, C, -10)$. So, A-C is found first but not A-B-C. Dijkstra assume that the minimality will never change when "closing" a node. The use of negative numbers change this rule and so is not compatible with Dijkstra.

(b) *Dijkstra's algorithm modifications*

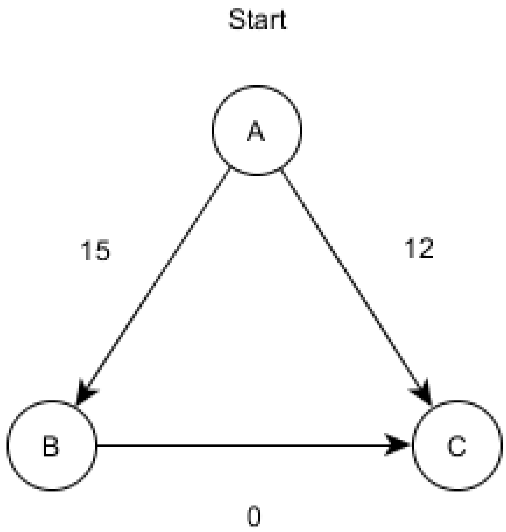

Figure 2: Graph with edge cost ≥ 0.

We can see that adding 10 to all weights doesn't work. A-C is still the path that Dijkstra find first and in this case, it is the shortest path which was not the case before.

(c) *Could BFS be used to find the shortest paths between nodes? If so, what would the edge weights have to look like for BFS to be guaranteed to find the shortest paths between nodes? Why (not)?*

With BFS, we can find a path with arbitrary weights but it is not guaranteed to be optimal.

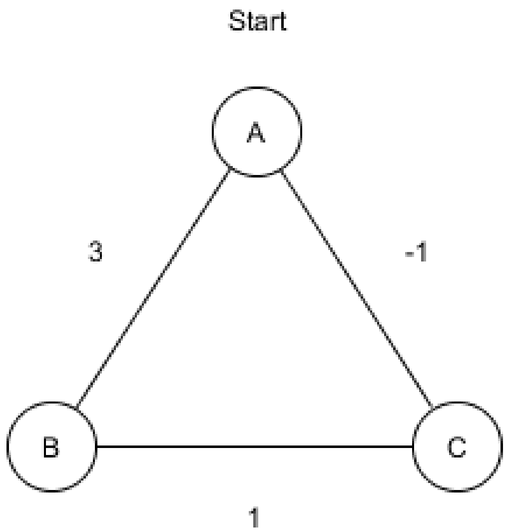

Figure 3: When we want to go from A to B, the shortest path is A - C - B, but with BFS we will first expand A and find the path A-B. When we expand the C node, B is already marked as "duplicate". BFS will ignore the link between B and C in this case.

Exercise 4.2: Force directed layout of networks

(a) *Implementation preparation*

For the harmonic force we have:

$$F_h(\vec{r}) = -\nabla E_h(\vec{r})$$

According to the definition:

$$= -\nabla \frac{k}{2} \|\vec{r}\|^2 = -\frac{k}{2} \nabla \|\vec{r}\|^2$$

$$= -\frac{k}{2} \nabla (\sqrt{x^2 + y^2 + z^2})^2$$

$$= -\frac{k}{2} \nabla (x^2 + y^2 + z^2)$$

$$= -\begin{pmatrix} kx \\ ky \\ kz \end{pmatrix}$$

For the Coulomb force we have:

$$F_c(\vec{r}) = -\nabla E_c(\vec{r})$$

$$= -\nabla (\frac{1}{4\pi\epsilon_0} \frac{q1q2}{\|\vec{r}\|})$$

$$= -\nabla (\frac{1}{4\pi\epsilon_0} \frac{q1q2}{\sqrt{x^2 + y^2 + z^2}})$$

$$= -\frac{q1q2}{4\pi\epsilon_0} \nabla (\frac{1}{\sqrt{x^2 + y^2 + z^2}})$$

We apply partial derivatives on $\frac{1}{\sqrt{x^2+y^2+z^2}}$ (chain rule):

$$= -\frac{q1q2}{4\pi\epsilon_0} \begin{pmatrix} -\frac{x}{(x^2+y^2+z^2)^{\frac{3}{2}}} \\ -\frac{y}{(x^2+y^2+z^2)^{\frac{3}{2}}} \\ -\frac{z}{(x^2+y^2+z^2)^{\frac{3}{2}}} \end{pmatrix} = \frac{q1q2}{4\pi\epsilon_0} \begin{pmatrix} \frac{x}{(x^2+y^2+z^2)^{\frac{3}{2}}} \\ \frac{y}{(x^2+y^2+z^2)^{\frac{3}{2}}} \\ \frac{z}{(x^2+y^2+z^2)^{\frac{3}{2}}} \end{pmatrix}$$

(b) ***Adapting the energy equations for networks***

For the Harmonic force we have:

$$F_h(\vec{r}_{i,j}) = -\nabla E_h(\vec{r}_{i,j})$$

K is the spring constant, because it's constant we can drop it:

$$F_h(\vec{r}_{i,j}) = -\nabla \frac{1}{2} \|\vec{r}\|^2$$

$$-\frac{1}{2} \nabla \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right)^2 = -\frac{1}{2} \nabla (x_i - x_j) + (y_i - y_j)$$

We apply the partial derivatives:

$$= -\frac{1}{2}\begin{pmatrix} y_i - y_j \\ x_i - x_j \end{pmatrix}$$

OR :

$$= -\begin{pmatrix} x_i - y_j \\ y_i - y_j \end{pmatrix}$$

For the Coulomb force:

$$-k_1 k_2 \begin{pmatrix} \frac{x_i - x_j}{((x_i-x_j)^2+(y_i-y_j)^2)^{\frac{3}{2}}} \\ \frac{y_i - y_j}{((x_i-x_j)^2+(y_i-y_j)^2)^{\frac{3}{2}}} \end{pmatrix}$$

(c) ***Understanding the Coulomb and harmonic energy:*** *How does the Coulomb energy and harmonic energy change if the degree of both nodes is increased or decreased? What happens if the distance between two nodes is increased or decreased?*

(d) ***Understanding the forces: Why is the Coulomb force the repulsive force and the harmonic force the attractive force?***

(e) ***Implementing the force directed layout algorithm***

(f) ***Simulated annealing:*** *Explain why simulated annealing is a worthwhile optimisation principle in practice.*

(g) ***Applying the layout algorithms***

Exercise 4.3: Graph Modular Decomposition

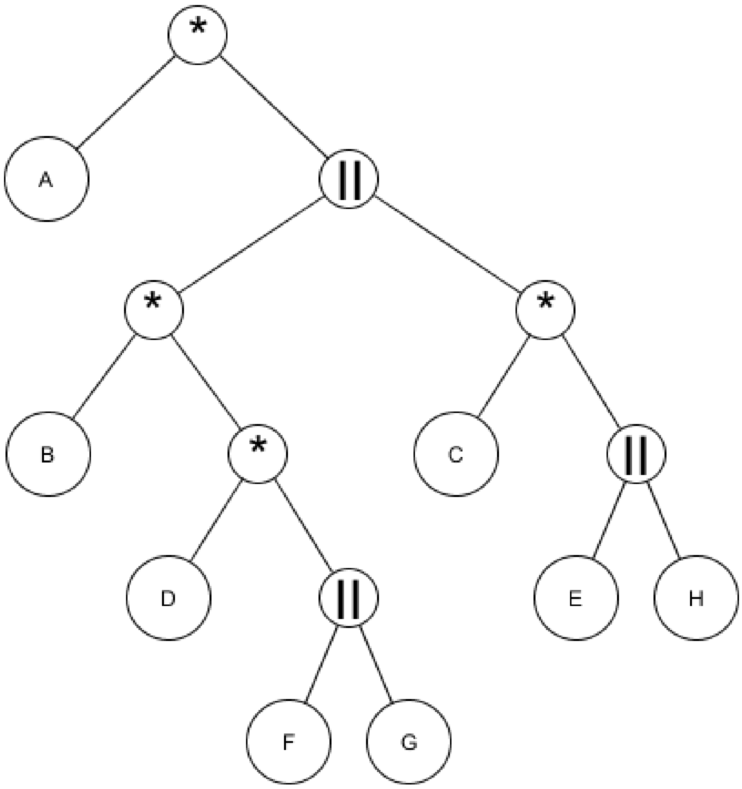

Figure 4: Modular decomposition of the network