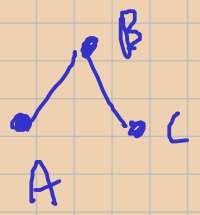
We will look at algorithms that maximize the modularity function

Let's talk about greedy algorithms: algorithms where at each step we try to make the most optimal choice!

We've seen the paper of _____ and Newmann, where we break links with high betweenness

There's a paper from Newmann alone from 2004, where there's an "agglomerative" approach

We start from a community



and we add one node from the ones that, in the edgelists, are connected to either A or B or C, Then we compute Q again, if the increase is positive, we keep the new node, if it's not then the new node will be the start of another community!

The Newman algorithm is not very used because it's computionally inefficient

- In the Blondel (2008) algorithm

The algorithm works in 2 phases

I phase:

Each node has his own community

We try to move node j in the community of node i, (if i and j are connected!), we do this for all possible nodes j that are linked to i, we compute Q each time and we choose to keep attached the node that provided the higher value of Q

II phase:

After doing the Ist phase some times over each community, we should have found some communities formed from more than one node:

We make a new network, where the communities are single NODES! We construct the edgelist and we put an edge between two nodes (they were two communities) if any of the original nodes inside the first one was connected to any of the original nodes inside the second one.

After doing an iteration of the second phase, we can do the Ist phase again

This algorithm is much more popular than Newman

- Those two algorithms seen use "greedy techniques" and are time consuming.

- There're some important algorithms that use "simulated annealing techniques".

  A probabilistic technique

  One exemple is the Guimerà-Amaral, not much used, but uses tools of statistical physics

- Some other techniques: "extremal optimization"

  It's used by Radatool! (remember to have radatool downloaded and ready for the examination!)

  They're done in an heuristic way (heuristic: not the most optimal way, but works)

- Conclusions, some problems with modularity algorithms:

1) Don't trust modularity! If you try to do algorithms based on modularity maximization, you will find communities even in E-R models (with one giant component) sometimes, where there should not be!

2) Sometimes, with small enough communities, modularity increases when we join them even if they are clearly separated communities

3) Modularity landscape: modularity is not a monotonic function, it has many MANY local maxima, and finding a "global" maxima is basically impossible

The solution to those kind of problems is "community characterization", done post algorithms