<u>**REPORT**</u>

The first thing to do was to create the graph where the nodes are our *authors* and the weight of edges is the *Jaccard distance*. Two authors are connected only if they share at least one publication. The result is a graph with 1,025,000 nodes.

Then we created a subgraph concerning the authors who published in a given conference at least once and, we computed the (normalized) centralities measures, also plotting them. We considered the conference 4634 obtaining our subgraph.
For degree centrality, we can have a look at the corresponding plot *(Fig.1)*. Note that the darker is the colour of the node, the higher is the value of this measure. The degree centrality tells us the number of publications an author shares in that conference with others, but we’re considering the relative measure so we can just assert for instance that this measure is equal to 0.03 in mean.

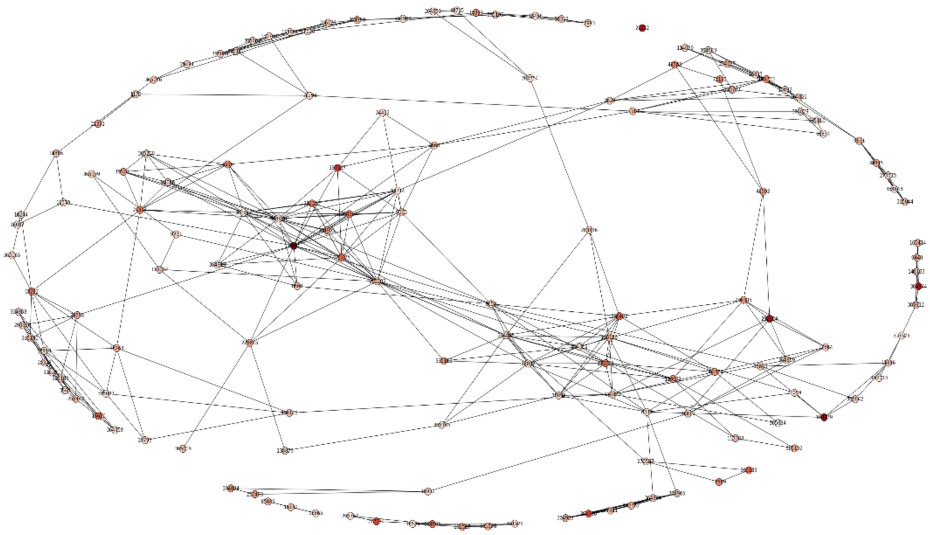

Fig.1

We also used the closeness centrality to measure how each node is close to the rest of the graph. Thus, the author with highest closeness centrality is the one with the higher number of shortest paths. Analyzing the *Fig.2* we can see that the nodes with a darker colour are the ones with higher closeness centrality, which are quite a lot. The mean of this measure is about 0.143.

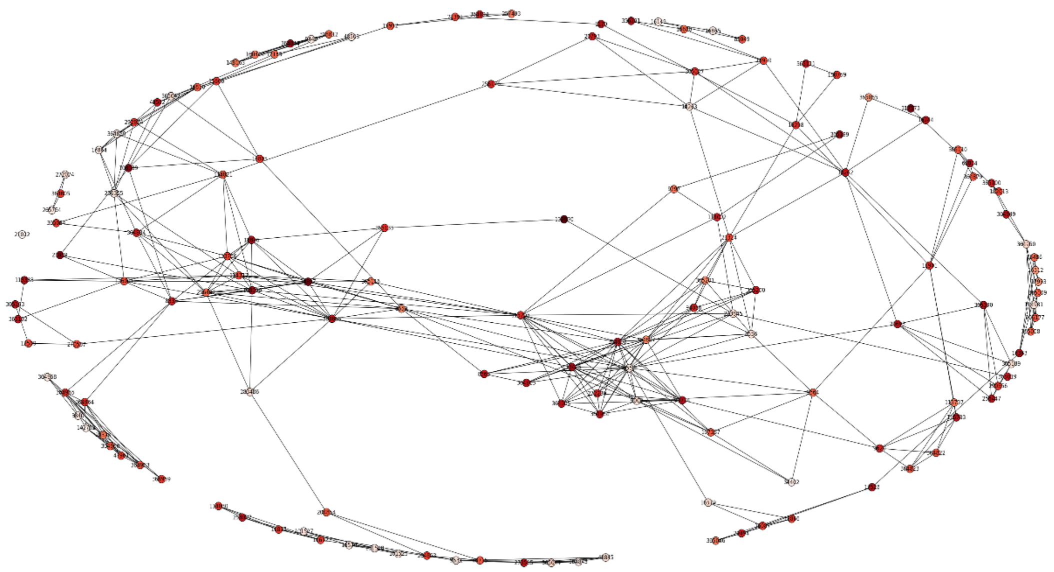

Fig.2

The last centrality measure we evaluated is the betweenness centrality, which is how a node can affect the communication between other nodes. In our case we can notice that there are few authors with high betweness centrality, in fact the mean is about 0.015, so there are no evident diversification of nodes (meaning clusters) in this graph, it is quite "homogeneous". As before, we plotted the result (*Fig.3*) so that we can see clearly the authors with higher betweenness centrality darker coloured.

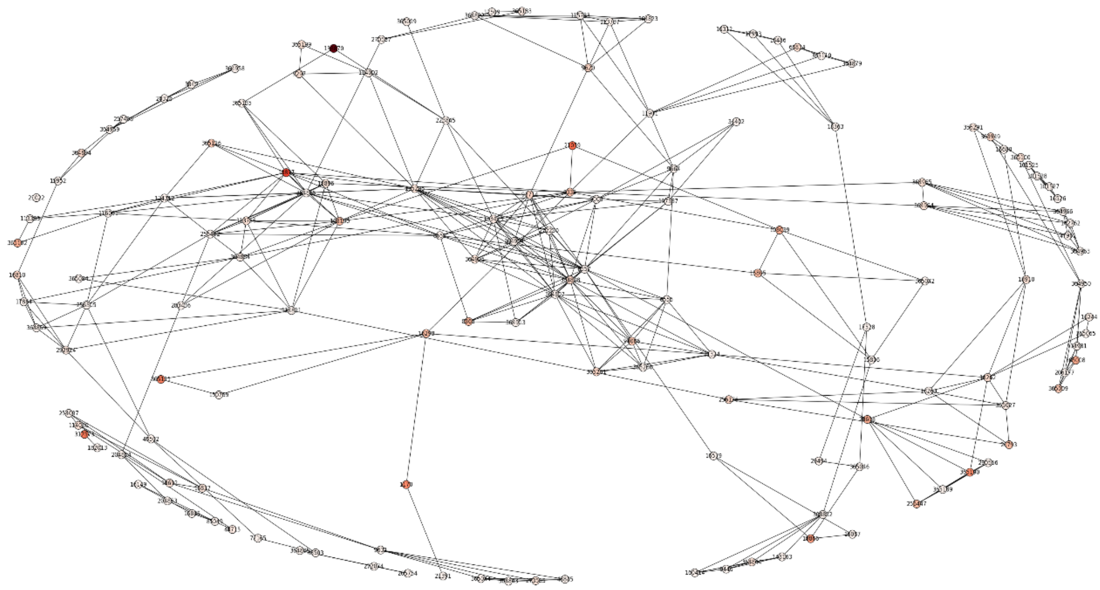

Fig.3

Next step was to take into account an author and an integer *d* representing the hop distance to find the subgraph induced by all nodes with hop distance at most equal to *d* with the author considered. In *Fig.4* is represented the resulting graph, considering *d* = 2. Even if *d* is a small number in this case, we obtained a very big network, so it's evident that there are many many links between nodes.

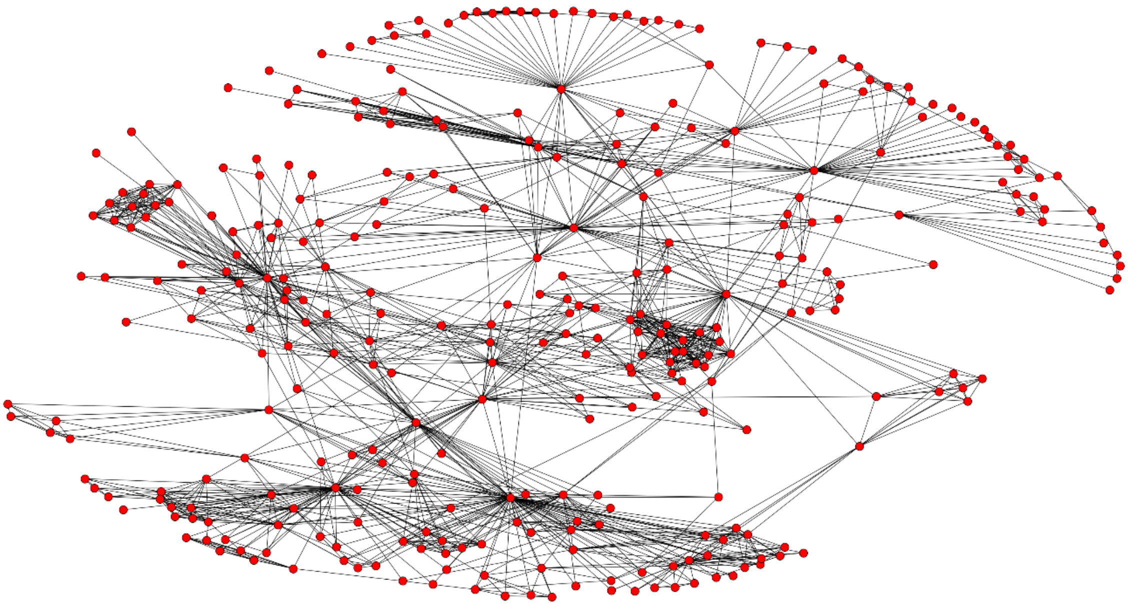

Fig.4

In the end we worked on the shortest path problem.

First, we took in input an author which is 16617 and computed the shortest path between him and Aris (256176), obtaining as result a shortest path with weight: 0.947.

Finally we considered in input a set of authors, computed the shortest path between each of them and the rest of the nodes in the graph and then took the minimum.

We started from these nodes {143709, 205236, 2067} and obtained these results:

- Nodes of 143709's shortest path : 143709, 216510, 232204, 566322, 216512, 88867, 143752, 435525
- Nodes of 205236's shortest path : 205236, 205388
- Nodes of 2067's shortest path : 178852, 123405, 523880, 11538, 2068, 2067.