

1 Introduction

We have an algorithm where we decompose a network into multiple shells using k-shell decomposition algorithm. Then we take a walk on this network. We start from a node and then look at its neighbours and move to the neighbour having the maximum coreness value. The question we are asking is regarding the number of steps this algorithm will take to terminate.

2 Progress

If a node has a coreness i , then it should have at least i neighbours having a coreness of i . If we could show that a node is almost always connected to a node having higher coreness, our work is done. This we need to do for the shells outside core, because for the core nodes, they have no other node having a high coreness.

We can try to see how much time does it take for a node to leave its current shell and enter an inner shell.