CS170 Final Project Methodology Overview

Walter Wu, Michael Cho, Tony Nie, David Phan

We used random algorithms to find solutions to most of the instances. We used the Python library Networkx. We used Networkx to build all of our graphs, with children as having a node weight of two, and adults having a node weight of one. Building the graph takes O(V+E) time. In our first random algorithm, we repeatedly find a cycle in the graph, and choose that cycle to be in our solution, then delete all nodes in the cycle from the graph. The cycle was found using the Networkx algorithm find_path(source, target), where the source is the next node in the list of nodes remaining in our graph, and the target is a node with an edge to our source. Running this on a given graph multiple times will produce the same solution, so we scramble the graph nodes randomly keeping all edges connected. We store the information connecting the old graph to the new graph in a dictionary. An example of how this might work is that given G ={V = [0,1,2}, E = [(0,1), (1,2)]}, G' could be {V = [1,2,0}, E = [(1,2), (2,0)]}. We run this as many times as we can on all instances. The second random algorithm corrects the problem that in the first algorithm, all cycles found will be minimum length cycles. In the second algorithm, once we find a cycle with length less than five, we remove a random edge in that cycle with a 50% chance. This new algorithm enables us to find larger cycles, albeit with a small chance and longer runtime.