CS170– Fall 2018— Solver Design Doc

Freddy Cervantes: SID 3033250924, Alex Kern: SID 27046342, Haitao Zhu: SID 3033251184

December 2, 2018

Stage 1: Rowdy Problem

The first step the algorithm does to solve the problem is take care of the rowdy kids. The algorithm does this in a set cover fashion. So given a list of rowdy groups. The algorithm iterates through the sets and finds student i with the most instances and sets him in the first bus. After, the algorithm iterates through the sets and removes all the sets containing i student. Now on the queue of sets to bee seen, we do not have any set with student i. We repeat the algorithm, finding the next student with the most instances and setting that student with the first student. By the end of this first stage, all the rowdiest students will be sat together in the first bus, or first set of buses depending on the the size capacity of the bus.

Stage 2: Optimizing Edges

Now that we have the rowdy children removed, the goal is optimizing the friend sets. Using the software "Mosek," a quadratic optimization model was formed maximizing the edges for each bus. Imagine a fully connected 3 vertices graph, such that we have $v_i \in v_1, v_2, v_3$. We say an edge exists in a subset if $v_i * v_j == 1$. this can only happen if $v_i == 1 \wedge v_j == 1$, else the edge will have been "removed". In the case of a fully connected 3 vertices graph the edges are accounted for by the equation $NumEdges = v_1 * v_2 + v_1 * v_3 + v_2 * v_3$. Note that in this case each vertices exists in the graph. Say we wanted to remove one of the edges, v_2 such that $v_1 == 1 \wedge v_2 == 0 \wedge v_3 == 1$. Then it is the case that NumEdges is equal to 1, and we have filled a bus of size 2. We use the edge-incident matrix H to accomplish this, by the following quadratic maximization equation:

$$Edge^* = \max_{v \in V} \; v^T H v + c^T v : \qquad s.t. \; v^T 1 = BusSize, \quad v_i \in 0, 1$$

Note that the problem is using $v_i \in 0, 1$, so given a large enough $|V|$, the software only makes a good prediction since it can not reasonably check $2^{|V|}$ possible vectors but gives a close to optimal solution.

What We Tried

We first just used a set-cover idea to remove the most edges from the rowdy groups. We then implemented the quadratic problem. There were several problems with the quadratic, first being we used a greedy approach for each bus. A better approach may have been to divide the graph in half, finding a way to do the min cut which would have used a divide and conquer approach. Unfortunately we all had many things going on so the approach we used was the only one we had time to implement.