CS170 Project Design Doc

Daniel Van Der Maden, Willow Watkins, Joseph Kraut

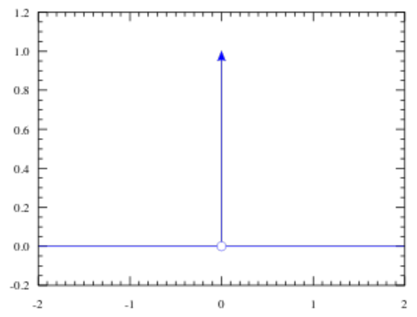
November 11, 2018

1 Greedy Dirac Delta Heuristic

Our method of a greedy algorithm involves the following heuristic

$$H(b,i) = \frac{f(b,i)+1}{\max_{g \in L, i \in g} \Phi(r(i,b,g))+1}$$

where $f(b,i)$ is the number of vertex i's friends currently on bus b, $r(b,g)$ is the number of members of rowdy group g on bus b. The $+1$ in numerator and denominator is for smoothing and continuity purposes. The function Φ is a function we design to fit the purpose of bringing down the heuristic as we approach a full rowdy group. A reasonable candidate for this is the Dirac Delta function shown below:



which has a definition of $\lim_{\sigma \to 0} N(0,\sigma^2)$, the limit of the normal distribution as the variance is taken to 0. This function, however, is not continuous and does not allow us the flexibility of penalizing additions that move a rowdy group close to completion. Instead we approximate this distribution with the normal distribution function, with sufficiently small variance σ^2, $\Phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-|G|)^2}{2\sigma}}$ where $|G|$ is the cardinality of rowdy group g. Of course, there are other reasonable heuristics we will try out such as exponentials, but this function has nice properties that make it an immediate candidate.

2 Optimization

After we achieve a solution with a greedy method, it is worthwhile to search the space for a solution to slightly increase score. We can do so in a number of ways, all of which we may try. First, and probably most practical, is a combinatorial optimization approach to make small steps in the optimization space to increase our solution's value monotonically. This however, can use up considerable time in computation as our optimization landscape is high dimensional (especially on large graphs with many choices of where to put students). To solve this we can try a Monte Carlo approach to speed up our combinatorial optimization. Another possibility to stack on top of the greedy approach is an intelligent optimizer such as Monte Carlo Tree Search or other methods involving branching. The basic framework for all these methods is the same, stack a more sophisticated optimizer on top of our greedy approach to squeeze slightly higher results out of our solution. The motivation behind this is that a well implemented optimizer will only increase our objective value and so we can only do better than or equal to the greedy approach alone.

3 Possible Other Approaches

We may also consider other approaches. One possible approach is a 2-step approach in which we greedily solve the problem first as if the rowdy groups didn't exist. We then correct for these rowdy groups in the solution possibly with dynamic programming, possibly greedily, etc. We may also try are k-Min-Cut approaches highlighted in the literature, time permitting. We were led to these possible solutions by a brief survey of literature and possible solutions to similar problems.