**Solving Capacitated Minimum Spanning Tree with a Genetic Algorithm**

**Section I: Introduction**

I will be creating a Genetic Algorithm to solve the problem of a Capacitated Minimum Spanning Tree(CMST). I find the CMST problem to be very interesting and I can see a few possible applications for its use. I have also done some exploration that will benefit my implementation of this algorithm.

**Section II: Problem Discussion**

CMST is basically a normal Minimum Spanning Tree(MST) except with two extra restrictions. The first restriction is that a specific node *r* will be considered the root of the tree. The second constraint is that no subtree can have more than *c* nodes. Nodes may also have a weight where *c* can then be defined as the sum of the weight of the nodes of a subtree can not exceed *c*.

I will be using a basic Genetic Algorithm to generate the solution to the CMST. I will compare the result I get against an excellent heuristic for the problem called the Esau-Williams Heuristic(EWH) which finds suboptimal solutions that are very close to exact solutions. EWH is the best heuristic I could find for the problem of CMST so I will be using that as the baseline to determine the efficacy of my implementation. If I have time at the end of implementing this basic Genetic Algorithm then I will try to add an additional operation that attempts to save sub-optimal, yet measurably different, genetic material to be reintroduced later to attempt to migrate peaks in a multimodal solution space.

**Section III: Reference Discussion**

The first paper I found is on the representation of individuals in a genetic algorithm as trees. Their goal was to create a representation that could not represent non-trees, could represent every tree with equal probability, and could easily be converted from representation to tree and vice versa. This will help me in my implementation because I wouldn’t want to have invalid solutions that I end up having wasted time processing because they aren’t even trees. Another paper that will help me with my implementation is [ENTER NEXT SOURCE HERE].

[Insert papers that give general background information about GAs][Probably use the Goldberg book as a reference, that helped a lot when I first learned GAs]

A paper I will be using for verification discusses the EWH algorithm and I will use that to better my understanding of that algorithm for when I need to implement that. I could create an algorithm that generates a solution to CMST, but there would be no use to it if I couldn’t tell whether or not the solution was any good or not.