**Technical Document**

*Zachary Petrusch, Nicklaus Benedict, Eric McAlpine, and Alex Lee*

- Parallel Explained (We can do this together tomorrow or Friday)

-Where it’s at, how it works

Starting with file I/O, we randomly created 100 parent routes. Then, each successive generation was created by taking 50 of the best routes and combining them to get 50 new routes. We used a merge sort to determine the best.

For the sorting of our parents, we use a parallelized merge sort due to its efficiency and ability to work with OpenMP. To start our project, we used a bubble sort to get started. With 1000 parents the bubble sort version ran in approximately 13 hours. With the implementation of a merge sort in serial, our run time was reduced to approximately 54 minutes. With OpenMP and parallel sections, our run time decreased further to \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Each successive generation would have the 50 best taken again so that the best of the previous generation are not lost by creating the children. We used a generation stopper appropriate to the size of the project to determine our stopping point by watching our best route. After running the merge enough times that no better route is found, we exit.

- How the Cross-Over/GA Work (Zach)

- Explain classes (Eric?)