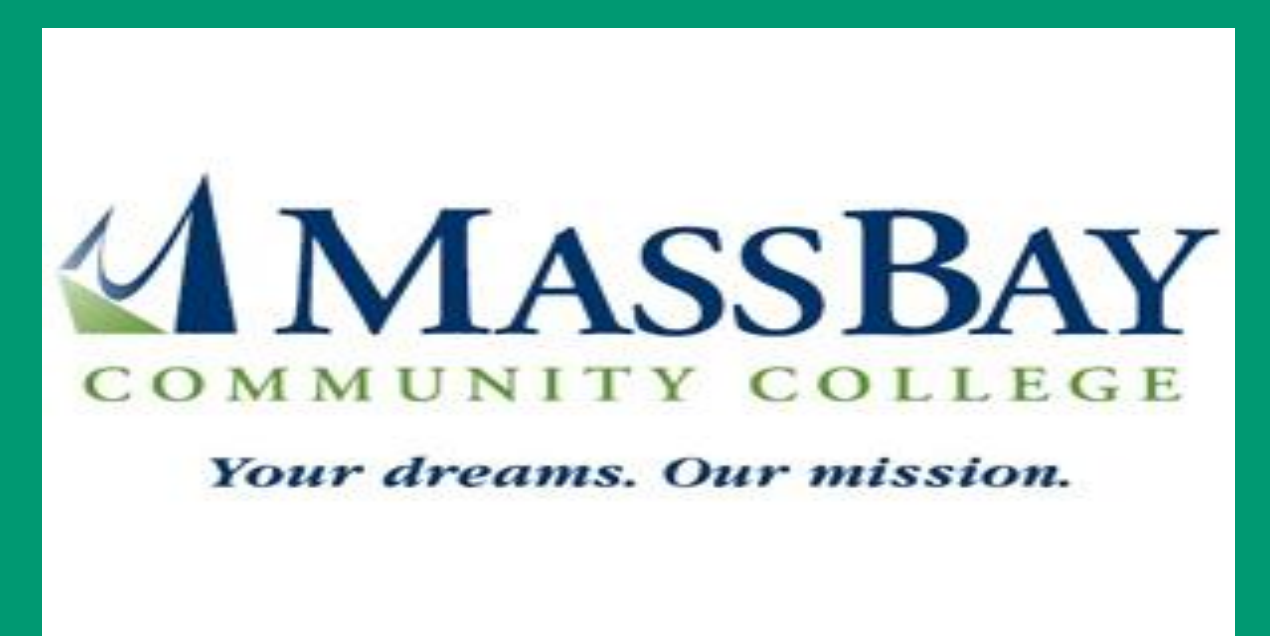




Finding the Optimal Route Around an Amusement Park to Maximize Fun and Minimize Time by Applying a Parallel Genetic Algorithm to the Traveling Salesman Problem



2015 Posters on the Hill

Washington, DC
April 22 – 23, 2015

MEGAN DELANEY
m_delaney2@post.massbay.edu

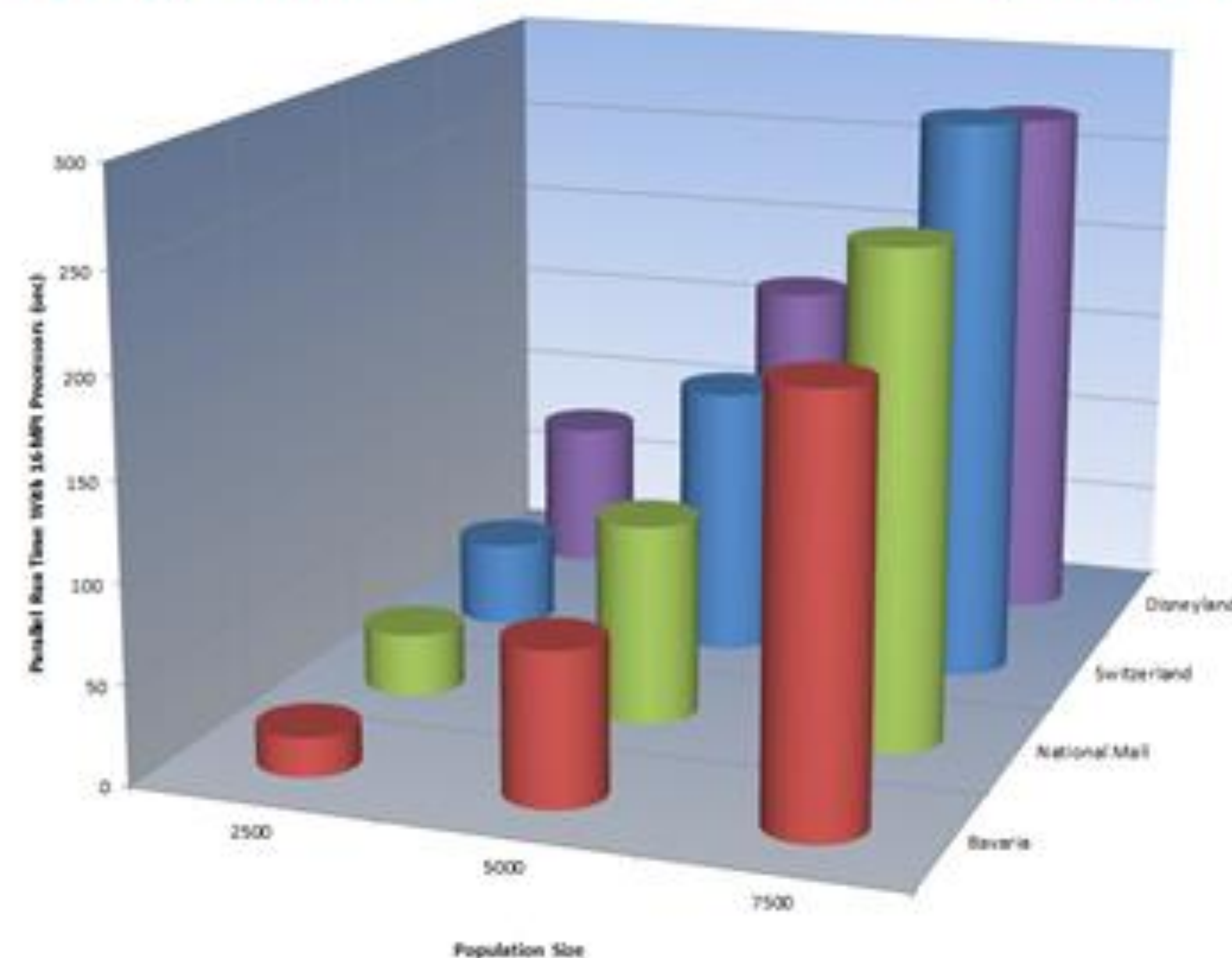
MASSACHUSETTS BAY
COMMUNITY COLLEGE

ADVISOR: GIUSEPPE SENA

SPONSOR: NATIONAL SCIENCE FOUNDATION (M-STEM GRANT)

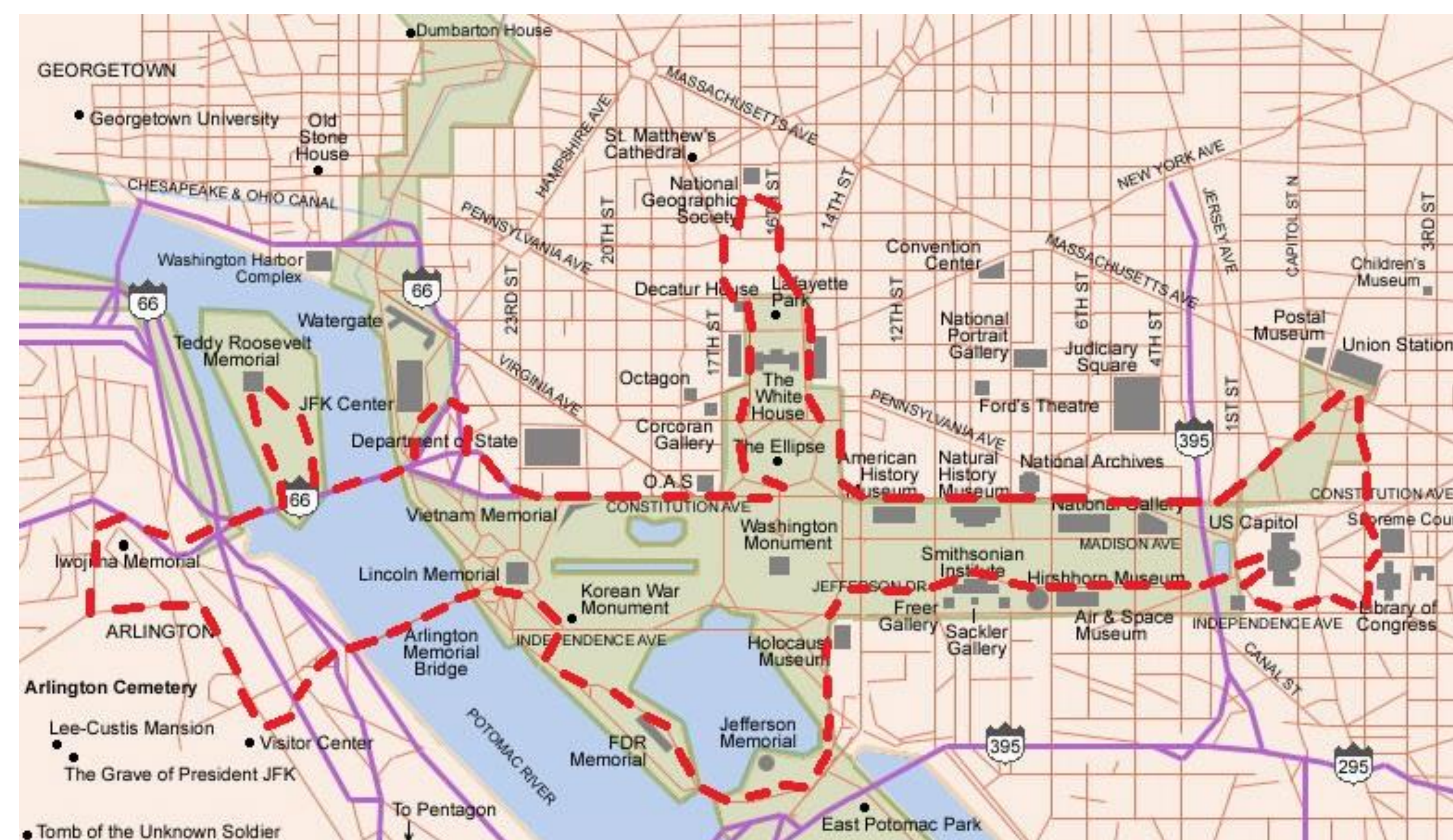
- While this graph shows selected data using 16 MPI processors, tests were run using different population sizes in a matrix of recorded tests.
- The results displayed here show the increased processing time with larger search spaces.

Processing Time with Different Population Sizes



An Efficient Tour

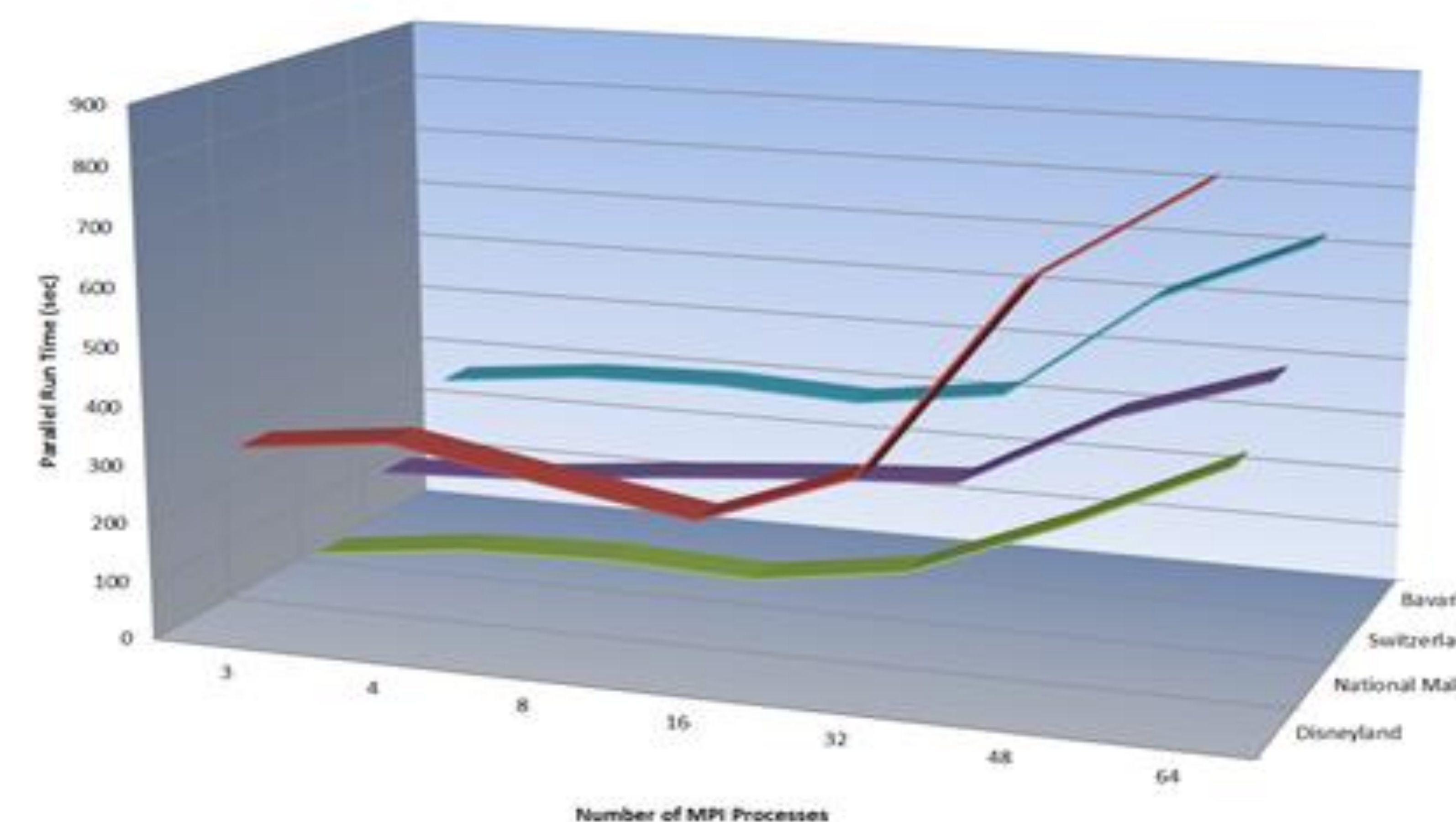
One of the data samples tested was a list of monuments and museums for an hypothetical tour of the National Mall in Washington D.C.



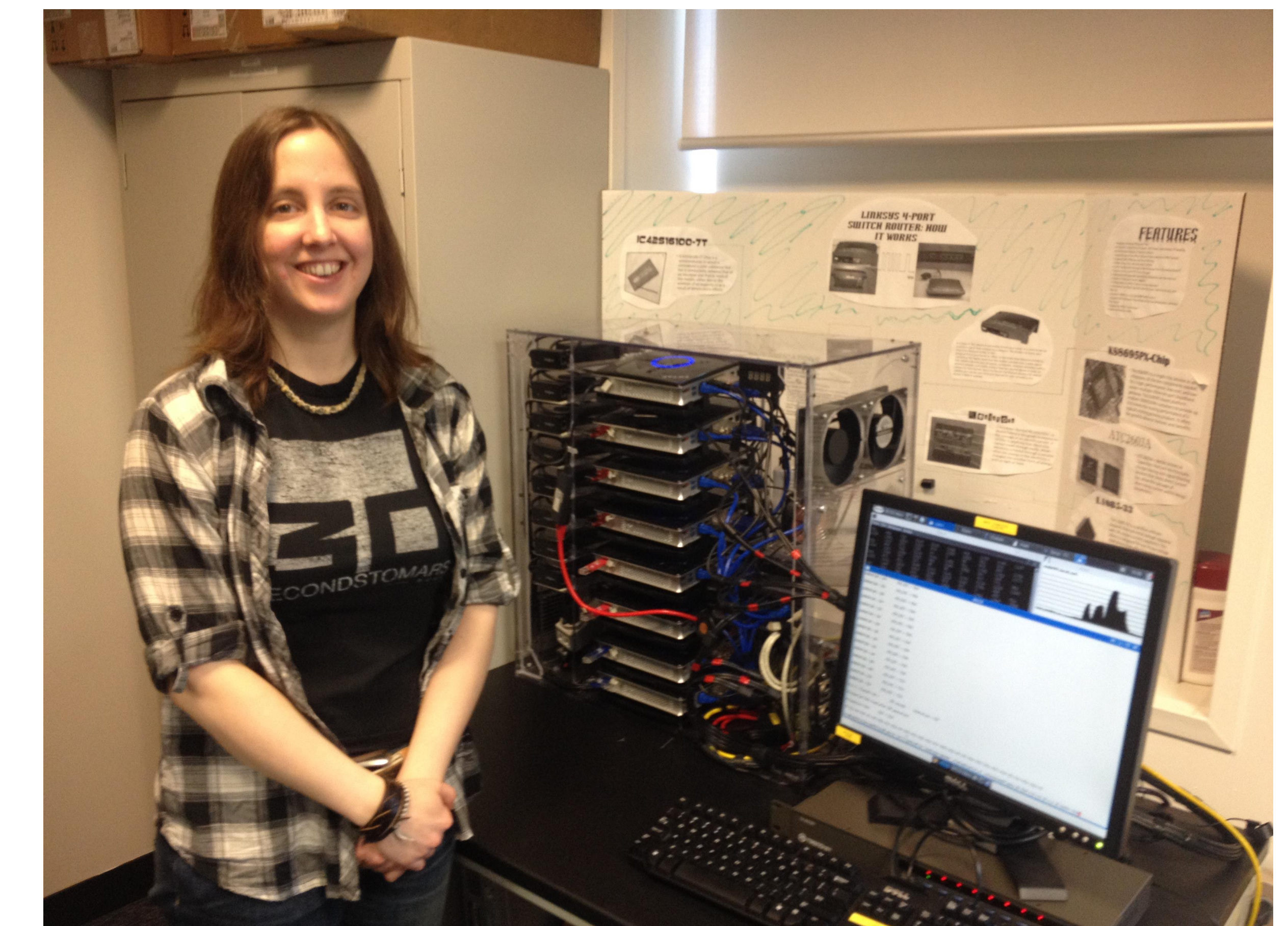
A Search For Answers

- The Traveling Salesman Problem is a type of optimization problem that can only be solved by an exhaustive search of possible solutions within the search space.
- A Genetic Algorithm, designed to mimic biological evolution, was used to solve the problem.
- This algorithm is implemented as a distributed application using the MPI (Message Passing Interface) library in order to expedite the search.
- Data samples were tested to measure the efficiency of the application.

Processing Time with Different Cluster Sizes



- There were many variables to consider when conducting optimization tests; the two main parameters that had the biggest impact on processing time are:
 - Number of MPI processors
 - Population of the search space.
- The algorithm's processing time increased when the distributed system's resources were pushed beyond the physical limitations.



The algorithm was tested on an 8-node Beowulf cluster dubbed *Cerberus*, using a cluster-based distribution of Linux with the MPI library installed.

The Happiest Path on Earth

The most efficient path through Disneyland Resort in Anaheim, CA



- Navigating complicated maps with multiple points of interest is just one of the uses for this algorithm.
- Other applications include: designing the layouts of power grids for infrastructure, optimizing routes for mass transit, emergency response vehicles, etc.
- Future work:
 - improve the efficiency of specific functions
 - add functionality to measure time (waiting and service) as an additional parameter
 - further analysis of the Genetic Algorithm parameters