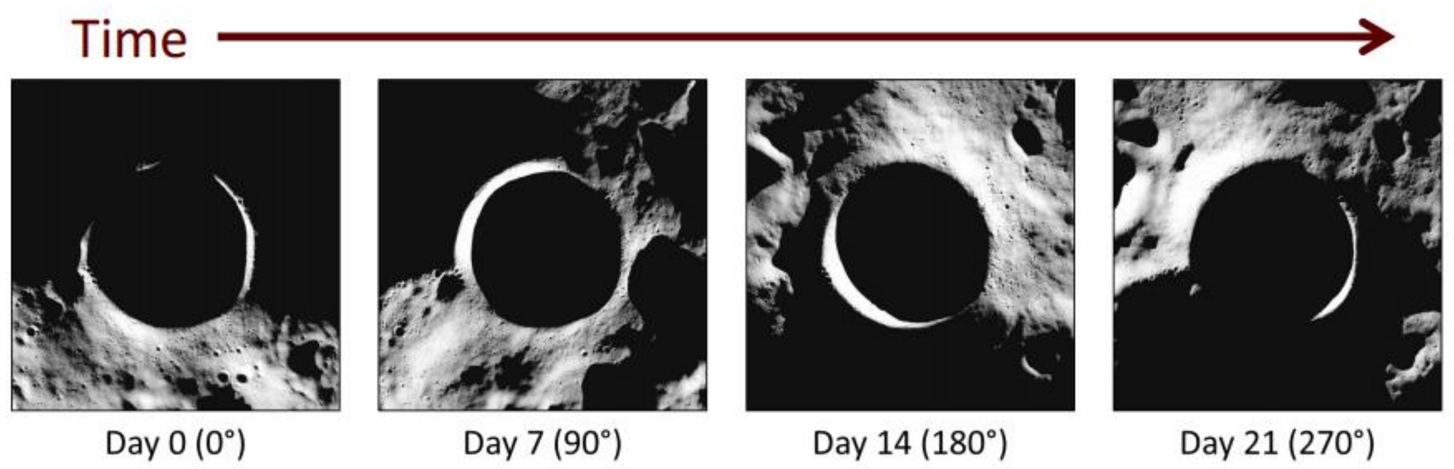
Waypoint Sequencing in a Dynamically Shadowed Environment

Jonathan Joo, *California Institute of Technology*

Advisor: Red Whittaker, *Robotics Institute - Carnegie Mellon University*

Introduction

- Planetary exploration often involves the visitation of a **sequence of waypoint destinations**. Planning determines which waypoints can be reached and in what order.
- Many regions of interest, such as the Lunar poles, exhibit **time-variant shadows** which intermittently illuminate or shadow waypoints.

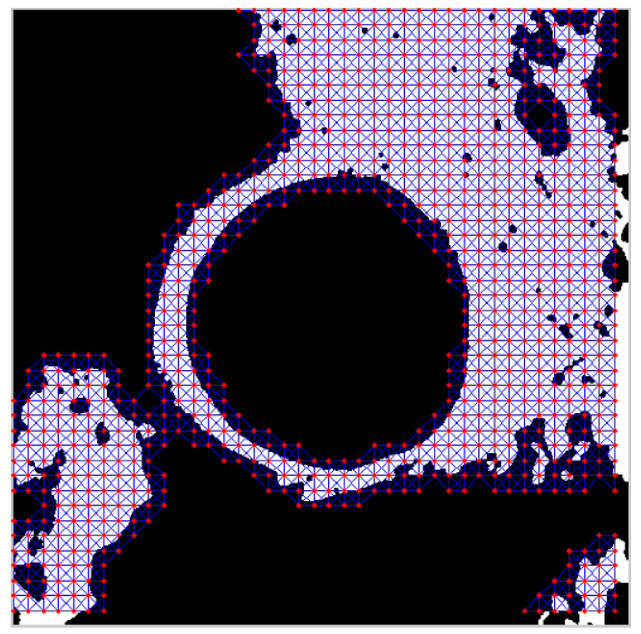


Shackleton Crater, Lunar South Pole

- Missions in these regions by solar-powered rovers thus require efficient planning to **multiple geographically distributed locations,** despite **time-varying solar illumination.**
- Thus, a waypoint sequencer was developed which **maximizes the value of locations visited** within a specified timeframe.
- Each waypoint contains an importance value, a time window for when it can be visited, as well as a "mission time" indicating how long a rover must stay at that waypoint before its value is counted.

Waypoint Sequencing Methodology

- Shadow maps of an environment over a time period are used to generate low-resolution graphs using **an 8-connected decomposition**. These shadow maps feature changing lighting/shadow conditions.

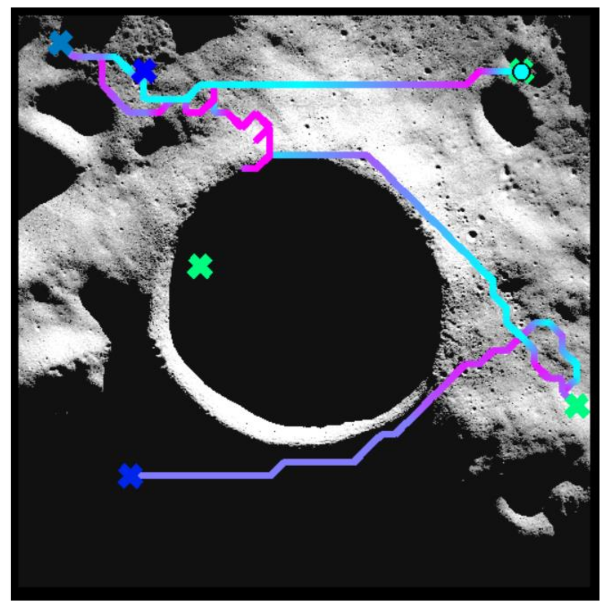


- Different **sequences of waypoints** are generated using various algorithms. Distances and travel times between waypoints are evaluated with an **A* planner**.
- Only feasible paths are returned from the sequencer, taking into account the rover's energy usage and time spent in shadow. If some waypoints are unreachable, they will not be included in the final sequence.

Brute force, greedy, and genetic algorithms were developed to address this problem.

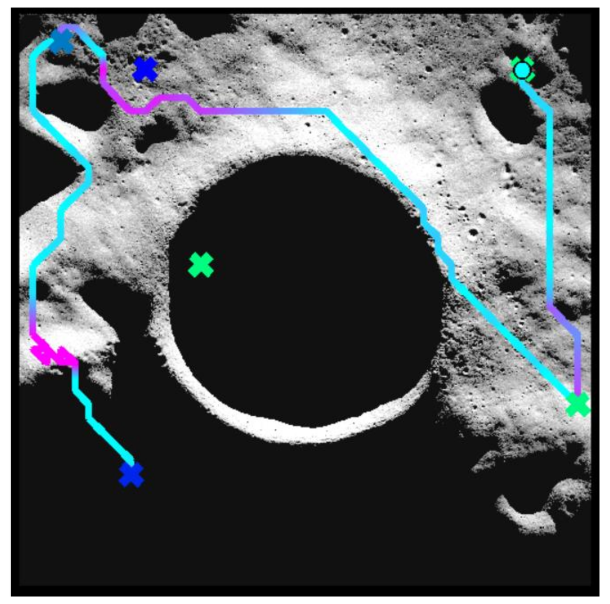
Brute Force Algorithm

Brute force looks at every possible sequence of paths that could be generated from the given waypoints, and chooses the best one. Brute force offers the benefit of being guaranteed to generate the optimal route, but it quickly becomes intractable. Not practical for cases more than ~12 waypoints.

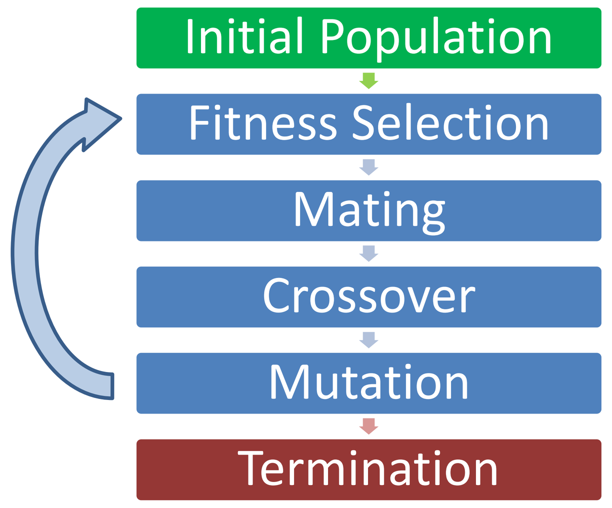


Greedy Algorithm

Greedy iteratively adds the next waypoint based on a function of value, distance, and time. Greedy offers the benefit of being very fast, even given hundreds of waypoints. However, it often returns suboptimal results, especially if points are organized in an unusual manner.



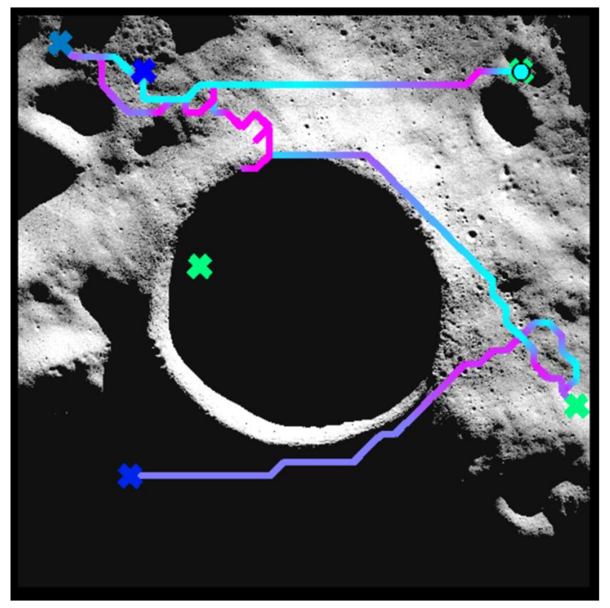
Genetic Algorithm



Genetic offers the benefit of guaranteeing more optimal paths than greedy, and at much faster speeds than brute given a large number of nodes.

It operates much like the biological process of evolution:

- An initial population is generated using combination of random and greedy algorithms.
- Fitness is determined based on total value of points visited.
- Mating uses a Queen-Bee process.
- Crossover uses Edge Recombination Crossover (ERC)
- Mutation is either addition of random waypoints within a sequence, swapping two random waypoints, or swapping a subset of waypoints.
- Termination occurs when the best sequence in a population is unchanged for 10 generations.



Results of Sequencing

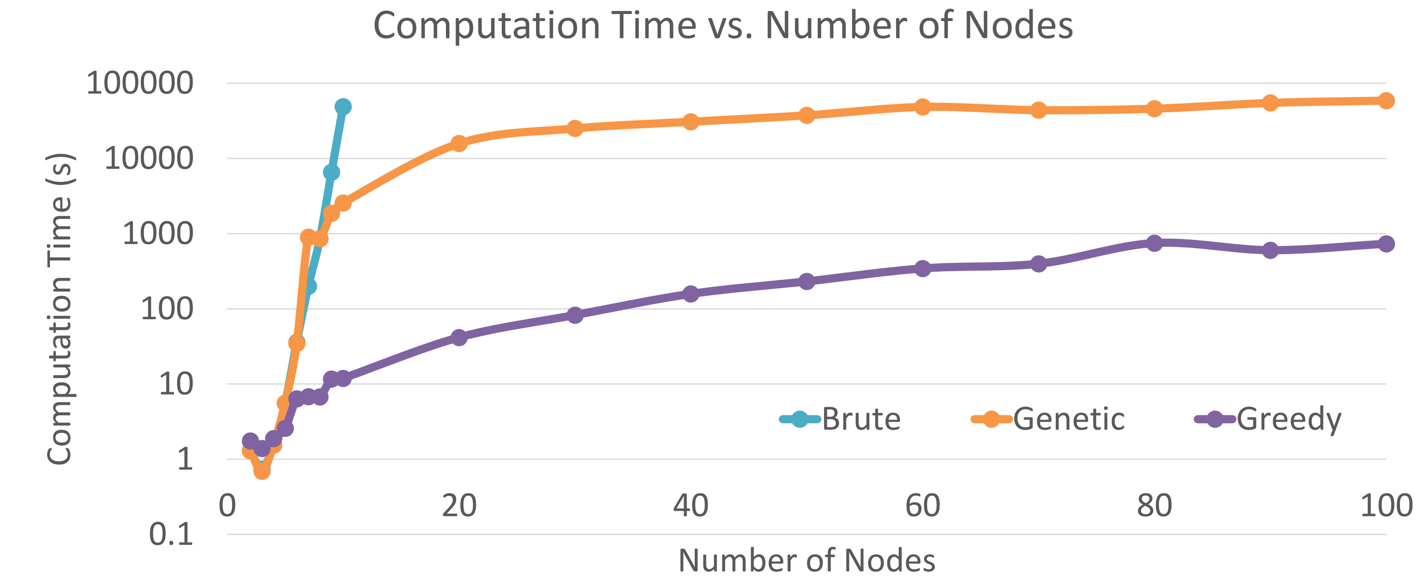


Figure 1. Log scale plot which compares the average running times of the various algorithms on different numbers of nodes.

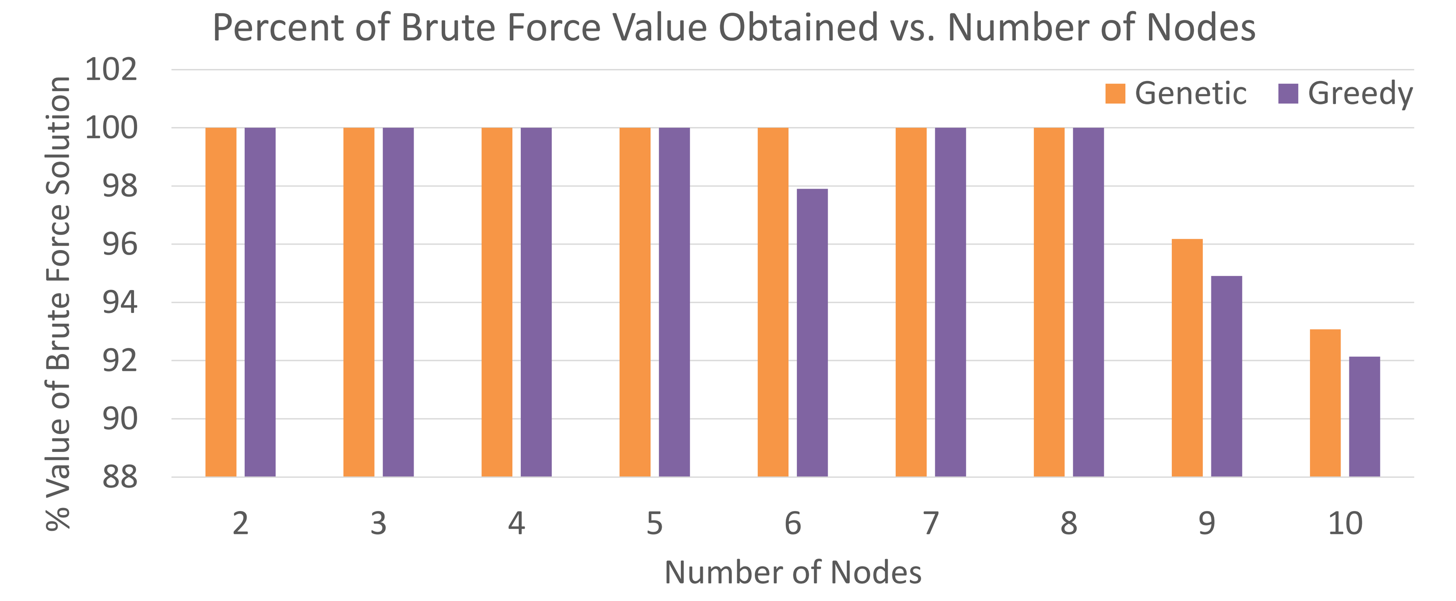


Figure 2. How the genetic and greedy algorithms compare to the optimal brute-force solution.

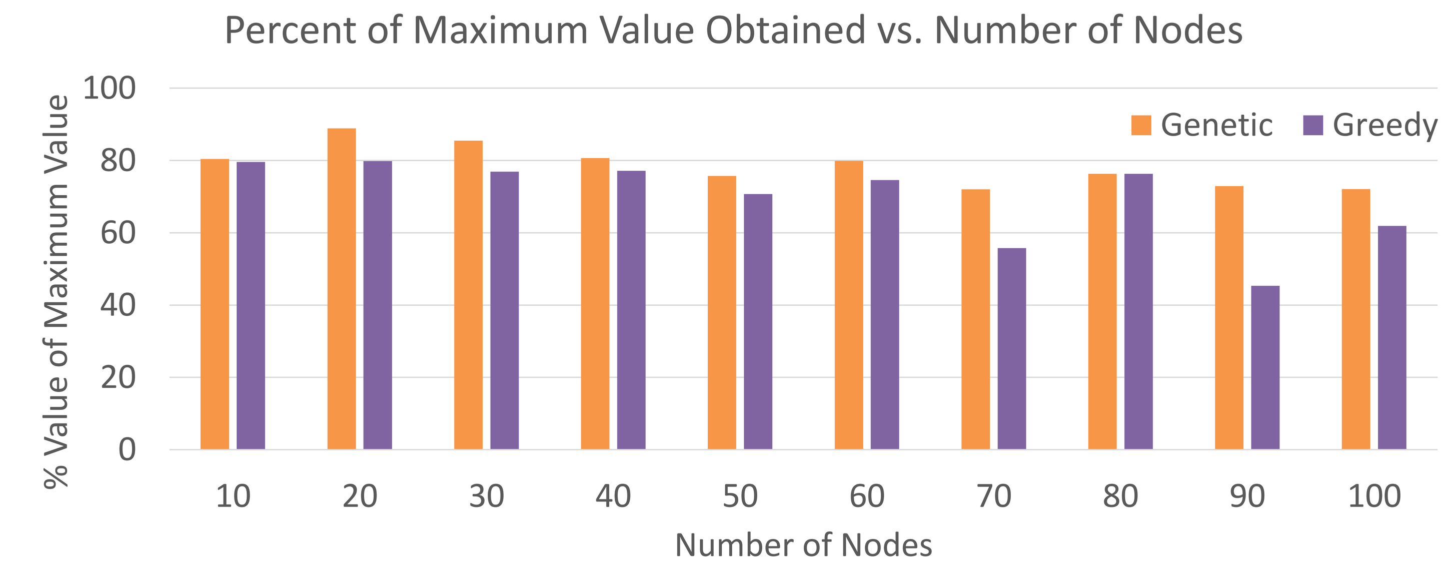


Figure 3. How the genetic algorithm compares to the greedy algorithm. Note that 100% is typically not achievable, even by brute force, since some waypoints cannot be visited.

Conclusions

Brute force, greedy, and genetic algorithms offer different merits for traversing numerous waypoints in time-varying environments. A good compromise between efficiency and accuracy is the genetic algorithm, which is significantly faster than brute force solutions, and generates higher-valued sequences than greedy solutions.

References

1. Tompkins, P. (2005). *Mission-Directed Path Planning for Planetary Rover Exploration*. Carnegie Mellon University.
2. Xia Wang, Bruce L. Golden, and Edward A. Wasil. Using a genetic algorithm to solve the generalized orienteering problem. In Bruce L. Golden, Edward Wasil, and S Raghavan, editors, The Vehicle Routing Problem: Latest Advances and New Challenges , pages 263-273. Springer, 2008.
3. Joanna Karbowska-Chilinska and Pawel Zabielski. Genetic Algorithm Solving the Orienteering Problem with Time Windows. In Jerzy Switek, Adam Grzech, Pawe Switek, and Jakub M. Tomczak, editors, Advances in Systems Science, volume 240 of Advances in Intelligent Systems and Computing, pages 609-619. Springer International Publishing, Cham, 2014.