Grid Search Algorithm for Set Optimization

Daniel Wasserman
Mentor: Zachary Feinstein

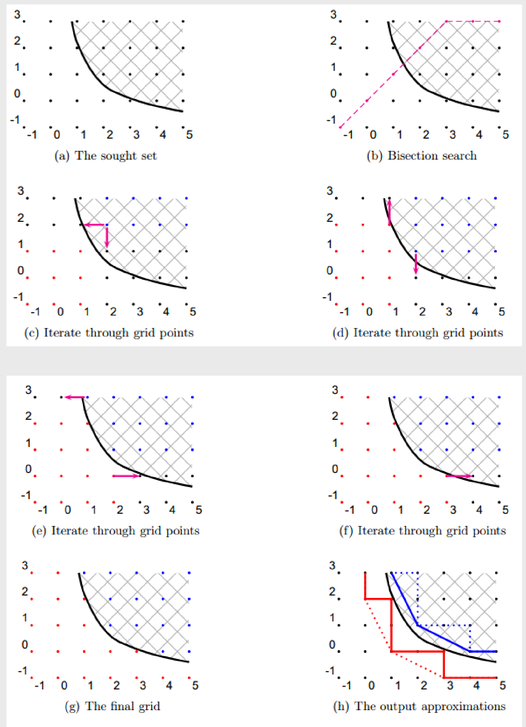
ABSTRACT

This project consists of developing an algorithm that performs a grid search with minimal computation and short run-time. The goal of a grid search is to approximate an unknown function that defines a set. The algorithm can test whether an individual point is inside or outside of the set. Using this information, the algorithm iterates through the grid points and determines which points are in the set. Professor Zachary Feinstein made MATLAB code that performs a grid search, although the program has a long run-time, especially when performed on grids in more than two dimensions. This drawback poses a need for an algorithm that selects points in a more intelligent way. Selecting fewer points can decrease run-time and improve efficiency. The improved algorithm utilizes MATLAB's parallel processing capabilities. The grid search starts on a sparse grid, then identifies regions close to the set boundary. The selected regions are then treated as separate grids. The grid search is performed on new grid points located inside each new grid. This process increases the number of total grid points, which allows for a closer approximation of the set's boundary. The structure of the improved algorithm allows the computer to take advantage of its multiple cores, in which regions are analyzed simultaneously. Run-time is shorter as a result. The process of dividing the grid into regions is repeated, recursively, until the desired accuracy is obtained. The algorithm then returns the boundary of the set, which approximates the unknown function.

Research Objective

- Overall goal: use MATLAB to approximate a function (in multiple dimensions)
- Points above the approximated function are inside the set, points below are outside
- We only have means of testing whether individual points are in the set or not

How Grid Search Approximation Works



My Work

- Understand and examine existing code developed by Professor Feinstein
- Rewrite the code myself, implementing different structural and mathematical techniques
- Learn how MATLAB's parallel processing toolbox works
- Use MATLAB's parallel processing toolbox to make a more intelligent algorithm that has a shorter run time, especially in more than two dimensions
- Test improved algorithm to measure its efficiency

Solution

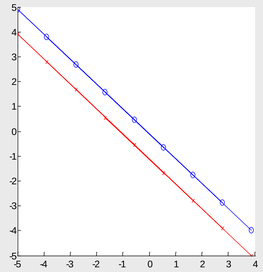
- Current algorithm structured to reduce time, although requires testing and debugging
- Starts with a sparse grid and determines upper and lower bounds
- Each lower bound paired with nearest diagonal upper bound to define a set of regions
- Regions are analyzed simultaneously, resulting in greater accuracy
- Recursive nature of algorithm repeats this process until desired accuracy is attained

Applications of a Grid Search Approximation

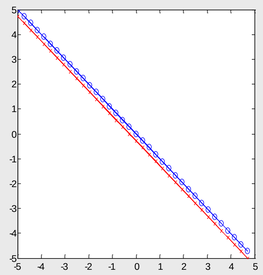
- Ideally, this code will provide ample approximations for solutions to systems under financial stress
- A mathematical way to model solutions of multidimensional sets that are too complex to visualize or construct
- Useful for general vector or set optimization

Visualization of Improved Algorithm

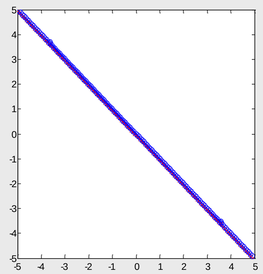
- This example approximates y = -x
- Extreme grid points are (-5,-5) and(5,5)
- Code runs through three iterations to achieve desired accuracy



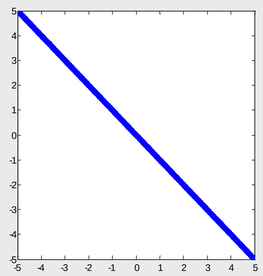
10x10 points. Each point on the lower bound defines a region to be analyzed.



The regions are divided into several 5x5 grids. The new approximation of the lower bound defines the next set of regions.



The process is repeated. 145x145 grid.



577x577 grid. Points above the line are inside the set, while points below are outside.