A Locally Convergent Search Technique Using a Delaunay Mesh Surrogate

Tyler Chang*, Layne Watson, Thomas Lux

Virginia Polytechnic Institute and State University

International Conference on Continuous Optimization
August, 2019

* corresponding author: thchang@vt.edu

VIRGINIA TECH

Table of Contents

Motivation

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

where f is an expensive black-box function, (i.e., a numerical simulation or real world experiment).

Motivation

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

where f is an expensive black-box function, (i.e., a numerical simulation or real world experiment).

Typical solution:

Motivation

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

where f is an expensive black-box function, (i.e., a numerical simulation or real world experiment).

Typical solution:

- Run global optimizer such as DIRECT search algorithm of Jones et al.

Motivation

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

where f is an expensive black-box function, (i.e., a numerical simulation or real world experiment).

Typical solution:

- Run global optimizer such as DIRECT search algorithm of Jones et al.
- For problems that are moderately smooth, DIRECT often converges quickly to the neighborhood of the solution, but painfully slowly to the exact solution

Motivation

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

where f is an expensive black-box function, (i.e., a numerical simulation or real world experiment).

Typical solution:

- Run global optimizer such as DIRECT search algorithm of Jones et al.
- For problems that are moderately smooth, DIRECT often converges quickly to the neighborhood of the solution, but painfully slowly to the exact solution
- Refine DIRECT solution using a "local" optimizer

Goal

Want the local optimizer to use DIRECT's database.

Statement of the Problem

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

- f is an expensive black-box function

Statement of the Problem

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

- f is an expensive black-box function
- given a finite dataset of n design points X in $\mathbb{R}^d$ and n corresponding objective values Y in $\mathbb{R}$

Statement of the Problem

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

- f is an expensive black-box function
- given a finite dataset of n design points X in $\mathbb{R}^d$ and n corresponding objective values Y in $\mathbb{R}$
- suspect that some pair (x, y) from the database is "in the neighborhood" of the optima

Statement of the Problem

$$\min_{x \in B} f(x) \quad \text{where } B \subset \mathbb{R}^d \text{ is simply bounded}$$

- f is an expensive black-box function
- given a finite dataset of n design points X in $\mathbb{R}^d$ and n corresponding objective values Y in $\mathbb{R}$
- suspect that some pair (x, y) from the database is "in the neighborhood" of the optima
- also $x^* = \arg\min_{x \in B} f(x)$ is in the convex hull of X.

The Mesh Refinement Scheme

- In iteration k, compute $DT(X^{(k-1)})$, where $X^{(k-1)}$ is the previous data set (let $X^{(0)} = X$) and $DT(X^{(k-1)})$ is a mesh

The Mesh Refinement Scheme

- In iteration k, compute $DT(X^{(k-1)})$, where $X^{(k-1)}$ is the previous data set (let $X^{(0)} = X$) and $DT(X^{(k-1)})$ is a mesh
- Judge the "fitness" of each element in the mesh $DT(X^{(k)})$ by the estimated objective value at its center

The Mesh Refinement Scheme

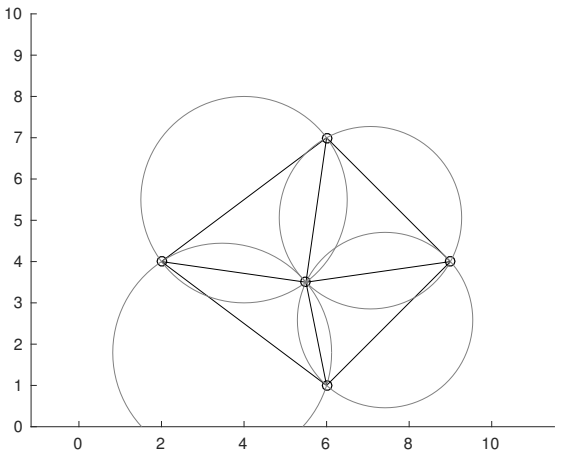
- In iteration k, compute $DT(X^{(k-1)})$, where $X^{(k-1)}$ is the previous data set (let $X^{(0)} = X$) and $DT(X^{(k-1)})$ is a mesh
- Judge the "fitness" of each element in the mesh $DT(X^{(k)})$ by the estimated objective value at its center
- Refine the mesh by evaluating $x^{(k)}$ at the center of element with the lowest estimated objective value

The Mesh Refinement Scheme

- In iteration k, compute $DT(X^{(k-1)})$, where $X^{(k-1)}$ is the previous data set (let $X^{(0)} = X$) and $DT(X^{(k-1)})$ is a mesh
- Judge the "fitness" of each element in the mesh $DT(X^{(k)})$ by the estimated objective value at its center
- Refine the mesh by evaluating $x^{(k)}$ at the center of element with the lowest estimated objective value
- $X^{(k)} = X^{(k-1)} \cup \{x^{(k)}\}$

What is the Delaunay Triangulation

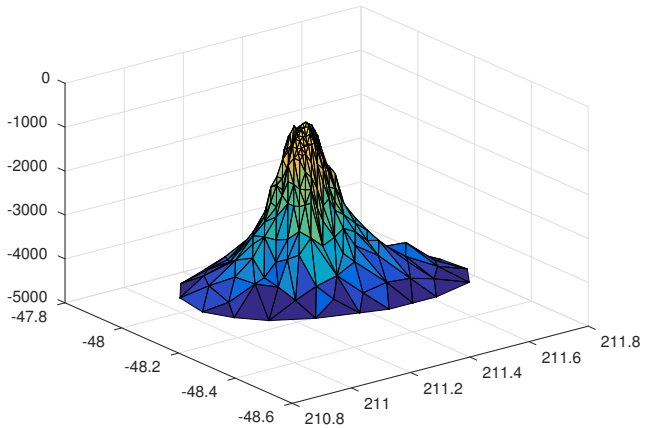
- Let P be a set of n points in $\mathbb{R}^d$
- Let $CH(P)$ denote the convex hull of P
- $DT(P)$ (the Delaunay triangulation of P) is an *unstructured simplicial mesh* over $CH(P)$, whose vertex set is P
- The *Delaunay triangulation* is a special triangulation, often considered optimal for interpolation and meshing purposes



VIRGINIA TECH

Uses

- Mesh generation (e.g., for finite element methods or computer graphics)
- Piecewise linear multivariate interpolation (e.g., for GIS)
- Topological data analysis
- Graph theory



VIRGINIA TECH

Linear Interpolation

- Let S be a simplex in $DT(P)$ with vertices $\{s_1, \ldots, s_{d+1}\}$
- Then $\{s_1, \ldots, s_{d+1}\}$ define a plane ℓ, along which we can linearly interpolate any $q \in S$
- Let $w_1, \ldots, w_{d+1}$ be *convex weights* such that $\sum_{i=1}^{d+1} s_i w_i = q$. Then

$$\hat{f}_S(q) = f(s_1)w_1 + f(s_2)w_2 + \ \ldots \ + f(s_{d+1})w_{d+1}$$

Linear Interpolation

- Let S be a simplex in $DT(P)$ with vertices $\{s_1, \ldots, s_{d+1}\}$
- Then $\{s_1, \ldots, s_{d+1}\}$ define a plane ℓ, along which we can linearly interpolate any $q \in S$
- Let $w_1, \ldots, w_{d+1}$ be *convex weights* such that $\sum_{i=1}^{d+1} s_i w_i = q$. Then

$$\hat{f}_S(q) = f(s_1)w_1 + f(s_2)w_2 + \ \ldots \ + f(s_{d+1})w_{d+1}$$

- $\hat{f}_S$ could also be used to extrapolate for $z \notin S$, akin to following the gradient estimate in S

Gradient Estimates

Owing to Lux et al. (citation pending), the following Lemma:

Let f be a function whose gradient is λ Lipschitz continuous in the 2-norm.

Let $\hat{g}_S$ be the (constant) gradient field in S with respect to $\hat{f}_S$

Then for any vertex s_j of S

$$\|\nabla f(s_j) - \hat{g}_S\|_2 \leq \sqrt{d}\frac{\lambda k^2}{\sigma_d}$$

where k is the maximum edge length of S and σ_d is the smallest singular value of $T = [s_2 - s_1 \quad \dots \quad s_{d+1} - s_1]$.

Takeaway: as the diameter of an element shrinks, its constant gradient estimate converges to the true gradient

VIRGINIA TECH

Method for Refinement

- For an element in the mesh, the objective value at its center is voted on using the gradient estimates of its $d+1$ neighbors (in this work, via averaging)
- The next function evaluation $x^{(k)}$ is chosen as the center with the most promising estimate

Convergence

- Within a neighborhood of the global minima, we would expect to see many design points sampled (requires proof)
- Then the estimated gradients converge, and the algorithm converges similarly to gradient descent

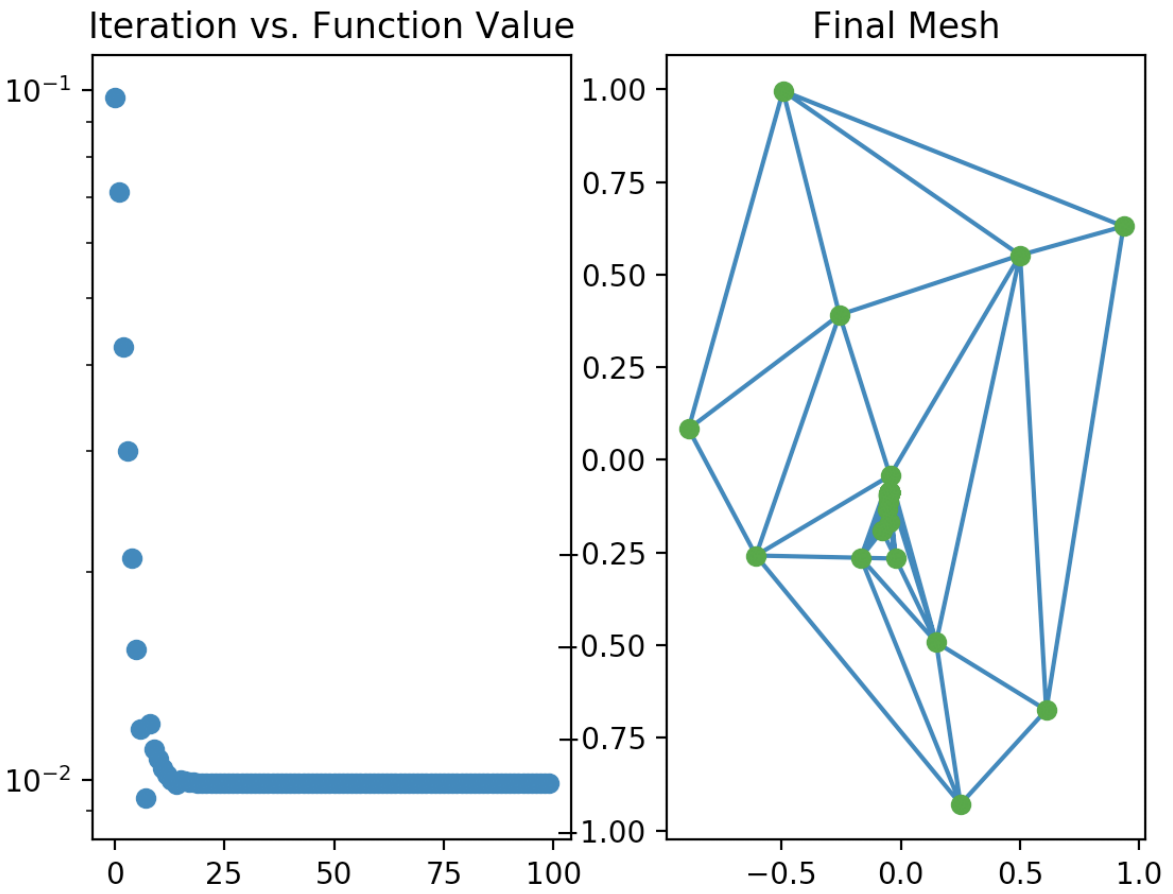
Experiments

The following preliminary experiments were run for 100 iterations over objective functions of two design variables using the Delaunay Search algorithm.

In all cases, the initial database was taken from a latin hypercube design of 20 points.

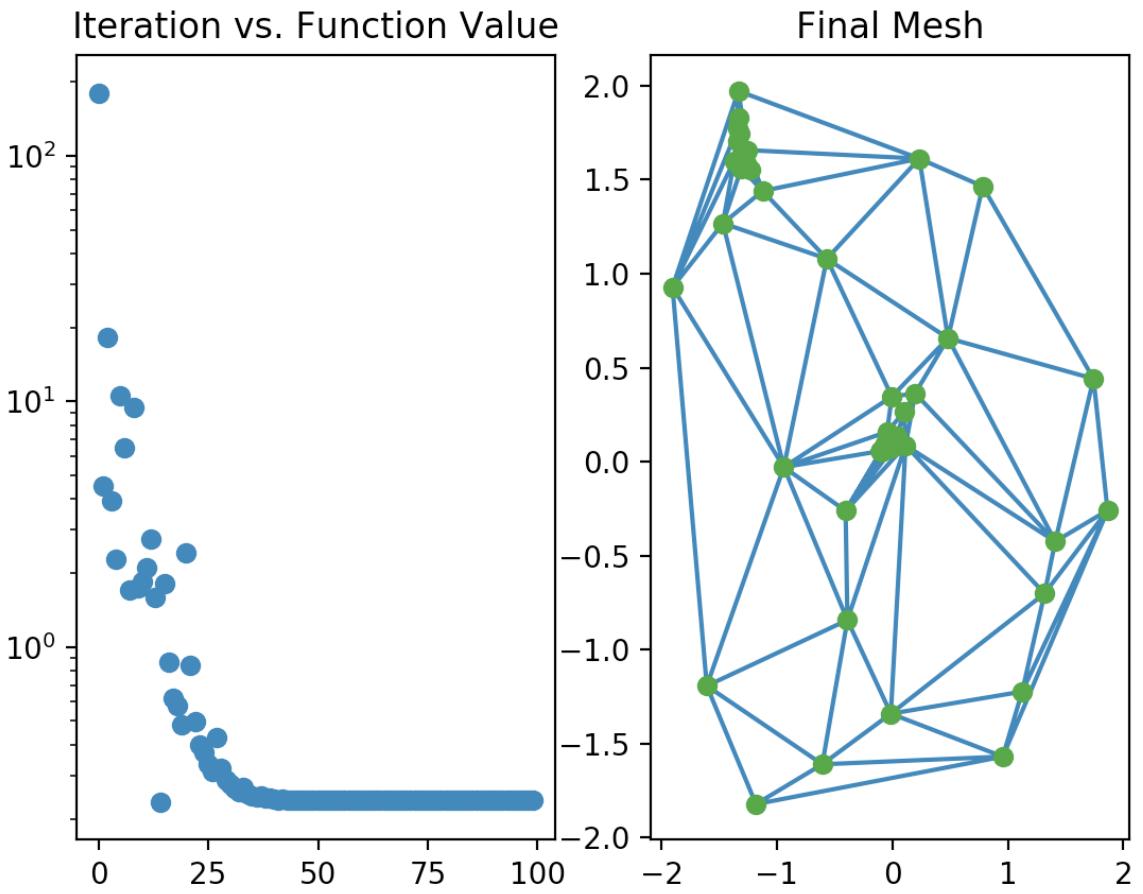
All functions were shifted so that their global minima has objective value $f(x^*) = 0$

Quadratic

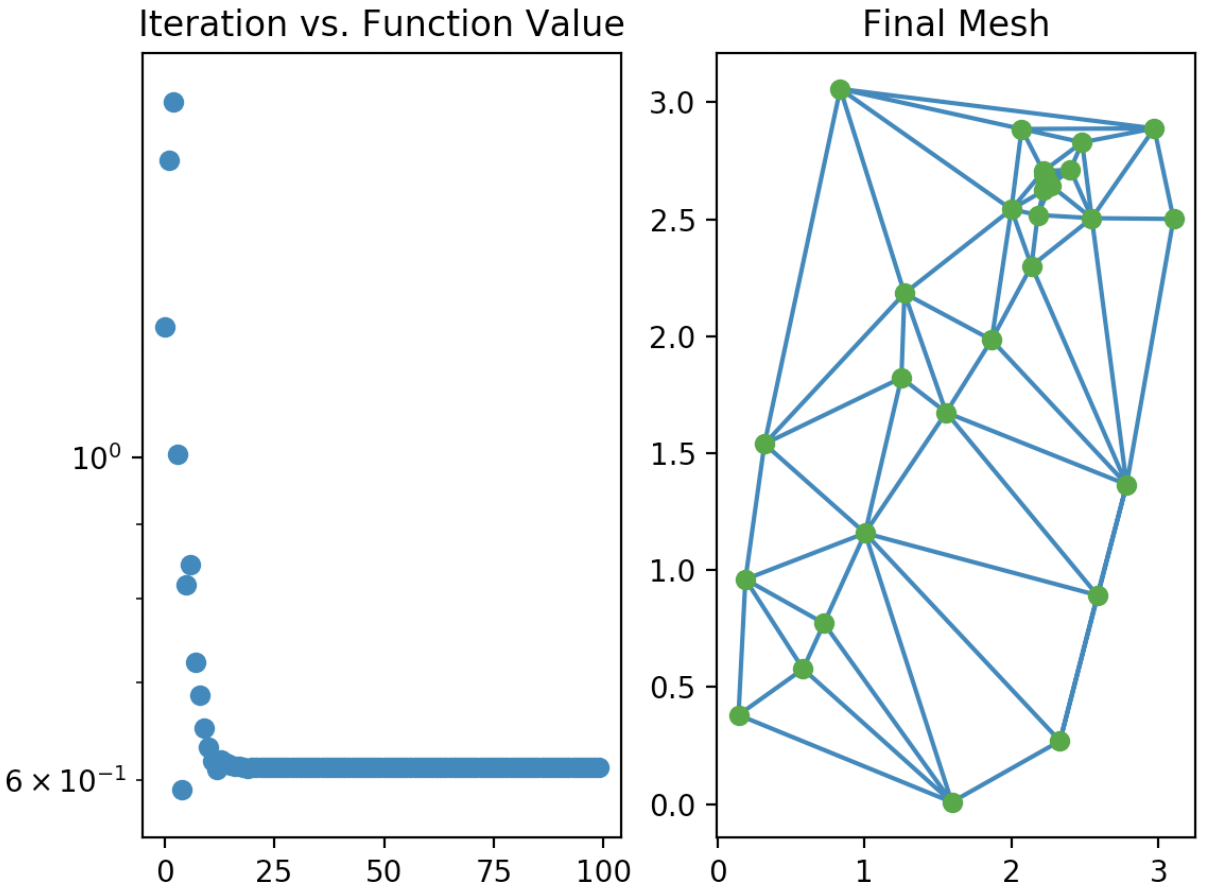


VIRGINIA TECH

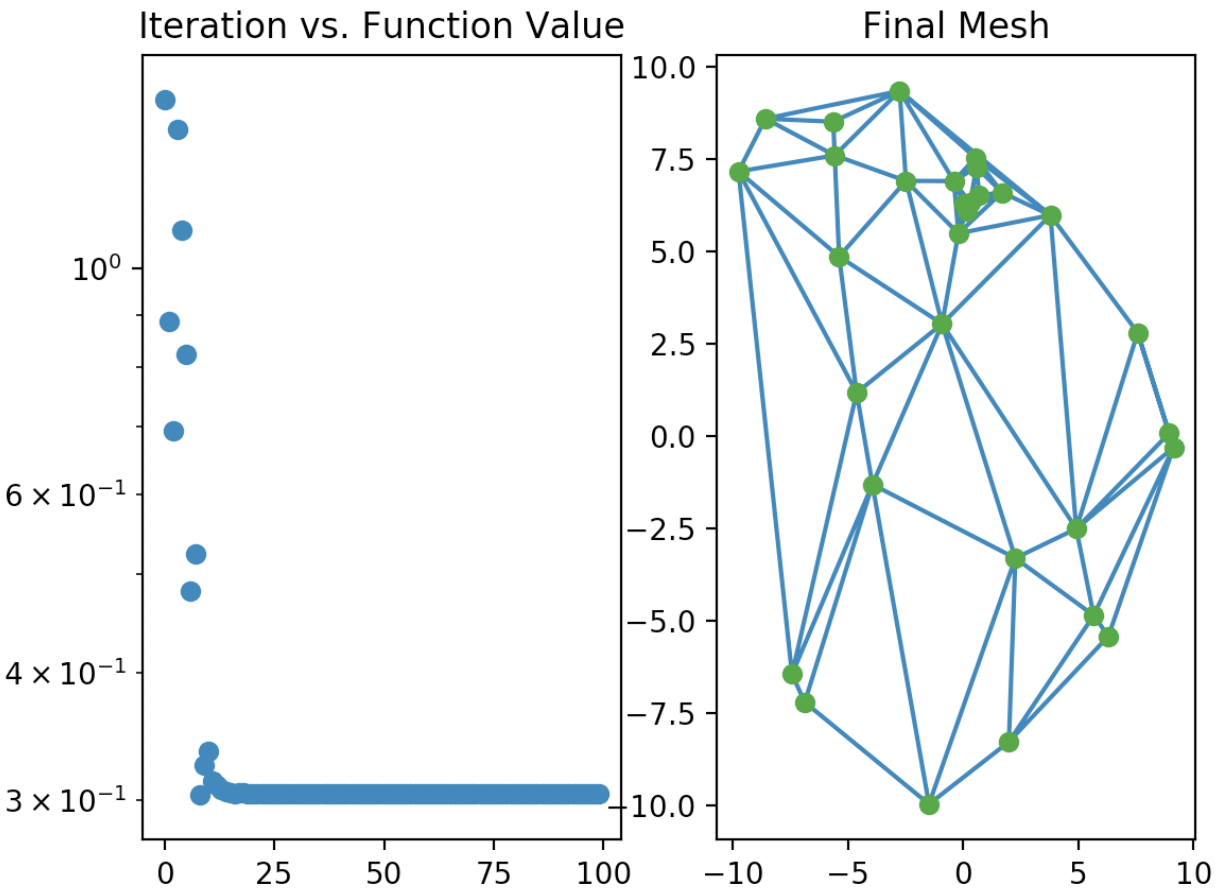
Rosenbrock


Iteration vs. Function Value
10^2
10^1
10^0
0
25
50
75
100
Final Mesh
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

Michaelwicz



Griewank



Limitations and Future Works

- Prove something about how data will be sampled in the neighborhood of the current observed minima (to show that the gradient estimates will converge in a neighborhood of a local minima)
- Currently, cannot sample outside the convex hull of the initial dataset X. Extensions for when $x^* \notin CH(X)$. Maybe LP?
- For d large, computing the entire Delaunay mesh is intractible. But using some properties of the Delaunay mesh and optimization problem, we should be able to eliminate many regions from consideration, making this tractible for relatively large d

QUESTIONS?