Feasible Points in Highly Constrained Simulation Optimization Problems

Caoheng Gu, Victor Krassovsky Mentored by Jeff Poskin, Joerg M. Gablonsky, Linhang Huang

Department of Mathematics, University of Washington – Seattle

DEPARTMENT OF MATHEMATICS
UNIVERSITY of WASHINGTON

Setup

Simulation-based Optimization:

- Necessary to model many physical processes
- Blackbox; often treated as a list of inputs to outputs
- Does not have nice properties necessary for more conventional optimization, e.g. not differentiable
- Expensive to query, querying too many times would be infeasible

Problems are usually highly constrained. Once we have a feasible point, there are techniques to optimize the objective value, the tricky part is finding the feasible point to begin with.
In this experiment, we define two example problems from the suite of test problems defined by Hock and Schittkowski.

Objective

We treat our optimization problem as a blackbox function that returns the total constraint violation for a given input. We want to find points that have 0 constraint violation.

Two main benchmarks:

- Feasible points per query
- Iterations until first feasible point

The techniques we present are applicable to either, our implementation focuses on the first.

Surrogate Model

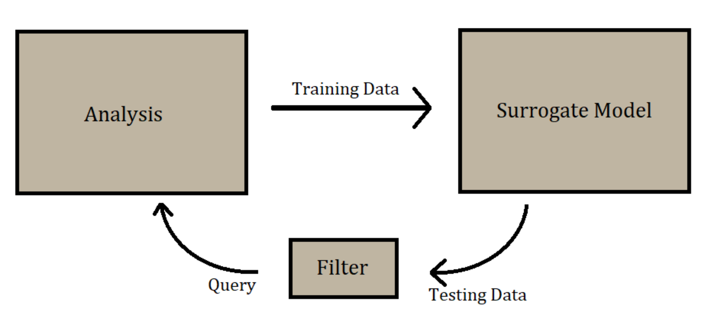

Figure 1. Surrogate Model Self-Reinforcment

A ***surrogate model*** serves as an efficient approximation of complex and expensive-to-evaluate functions. We first sample data points randomly and use them to construct the surrogate model. Then we iteratively train the surrogate model. During each iteration, we pass data sampled from the surrogate model through a filter which decides what data we train on during the next iteration.

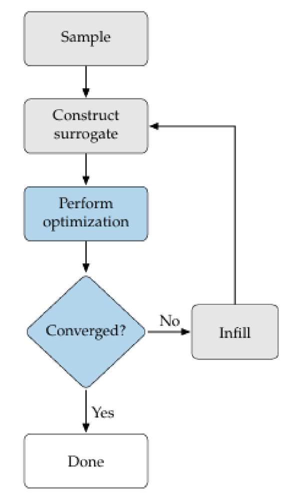

Figure 2. Process of Surrogate-based optimization[1]

Kriging Models/Gaussian Processes

Kriging implements a classic Gaussian Process Regression whose expression is:

$$\hat{y} = \sum_{i=1}^{k} \beta_i f_i(x) + Z(x)$$

$Z(x)$ is a realization of a stochastic process with mean zero and spatial covariance function given by :

$$cov[Z(x^{(i)}), Z(x^{(j)})] = \sigma^2 R(x^{(i)}, x^{(j)})$$

where σ^2 is process variance while R is the correlation.

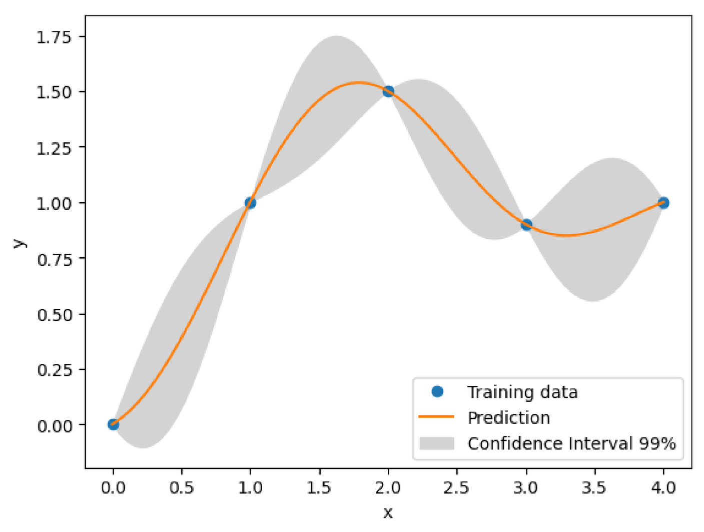

Figure 3. Kriging model[2]

Bayesian Optimization

The ***expected improvement*** (EI) acts as acquisition function to guide the search for the optimum. If f^* is the best solution we get from training, the improvement(I) is given as:

$$I(x) = \max(f^* - f(x), 0)$$

The ***expected improvement*** (EI) is given by:

$$EI(x) = \mathbb{E}\left[\max(f^* - f(x), 0)\right]$$

$$EI(x) = (f^* - \mu_f(x))\Phi\left(\frac{f^* - \mu_f(x))}{\sigma_f(x)}\right) + \sigma_f(x)\phi\left(\frac{f^* - \mu_f(x))}{\sigma_f(x)}\right), \Phi, \phi : CDF, PDFofN \sim (0,1)$$

- Train the Kriging model with sampled data points
- Compute the EI of every data point and sort
- Saving those with larger EI for further iteration

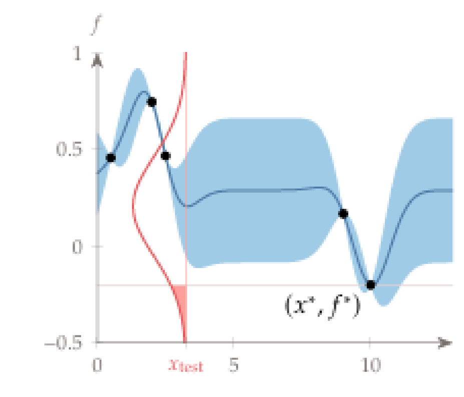

Figure 4. Expected improvement calculation[1]

Sampling & Selection

Our sampling method of choice is ***Latin Hypercube Sampling*** (LHS). It is psuedo-random and uniform over the the input-space.

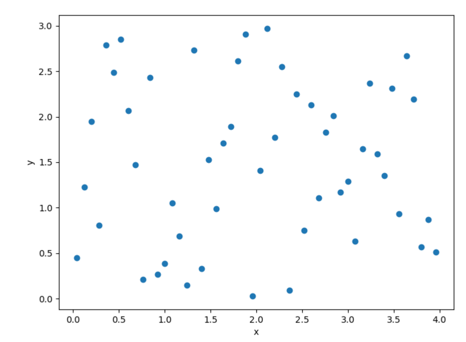

Figure 5. Latin Hypercube Sampling [2]

Additionally, during self-reinforcement, we filter points that are too close to get a sparser data set. Since our analysis is expensive, it's not desirable to query points that are too close.

Results

We were able to find feasible points at a much higher ratio then pure random sampling.

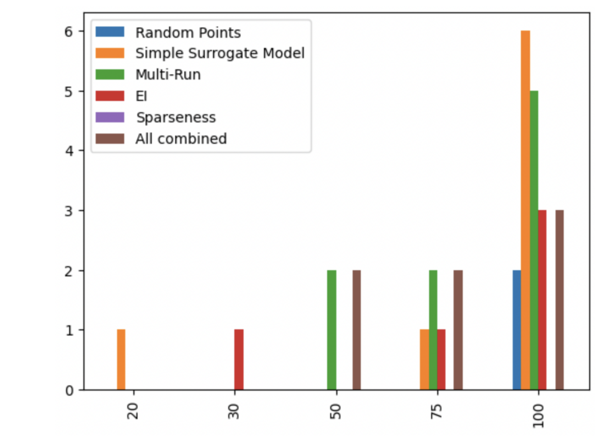

Figure 6. Benchmark of all experiments

It seems like the expected improvement + multi-run method works best for a limited number of queries while a basic surrogate model approach works well when many queries are allowed.

Next Steps

There are several things we have not yet implemented:

- Hyperparameter fine-tuning
- Wider problem-set
- Other surrogate models
- Other acquisition functions

References

[1] Joaquim R. R. A. Martins and Andrew Ning. *Engineering Design Optimization.* 2022.

[2] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier, and J. R. R. A. Martins. SMT 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables gaussian processes. *Advances in Engineering Sofware*, 188:103571, 2024.