Surrogate Modeling of Simulations for Multiobjective Optimization Applications

Tyler Chang and Stefan Wild

Mathematics and Computer Science Division
Argonne National Laboratory

July, 2021

Outline

Introduction and Motivation

- Notation
- The Simulation Function

PARMOO

- Python Structure
- Algorithmic Components

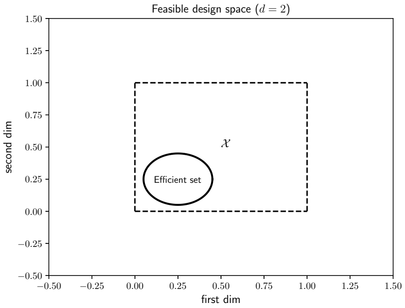
The Fayans EDF Problem

- Problem Description
- Theoretical Benefits

Experiments

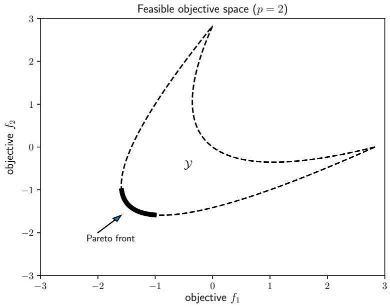
- Reduced Difficulty Problem
- Embedding Fayans EDF in PARMOO
- Results and Conclusions

Basic Notations and Terminology



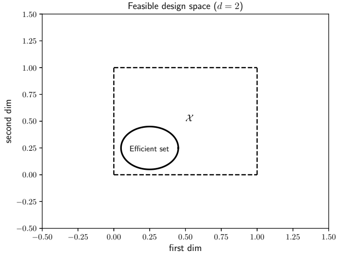
Design space

Objective Functions $\longrightarrow$

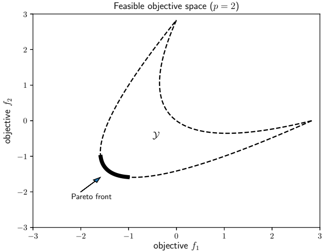


Objective space

Adding the Simulation



$$\xrightarrow{\text{Simulations}} \mathcal{S} \xrightarrow{\text{Objectives}}$$



Design space

Objective space

The Simulation Functions

The simulations are denoted by a vector-valued function

$$\mathbf{S} : \mathcal{X} \to \mathcal{S}.$$

Then the objective becomes

$$\mathbf{F} : \mathcal{X}, \mathcal{S} \to \mathcal{Y}, \qquad \min_{\mathbf{x} \in \mathcal{X}} \mathbf{F}(\mathbf{x}, \mathbf{S}(\mathbf{x}))$$

and the constraint becomes

$$\mathbf{G} : \mathcal{X}, \mathcal{S} \to \mathbb{R}^p, \qquad \mathbf{G}(\mathbf{x}, \mathbf{S}(\mathbf{x})) \leqq 0.$$

Opening the Blackbox

Our research question:

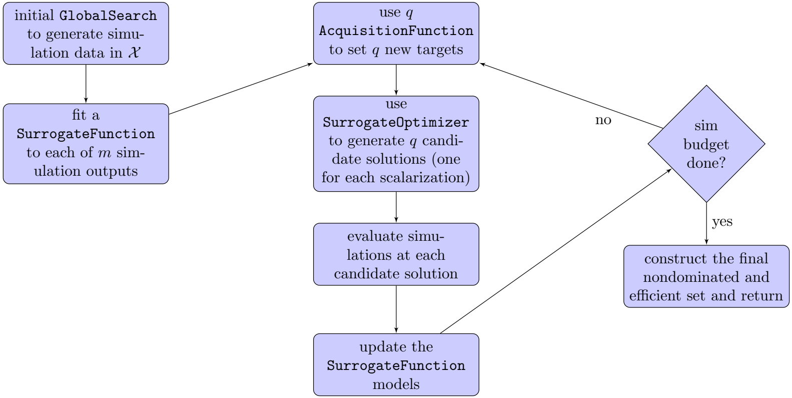
Can we take advantage of the simulation structure when the objective $\mathbf{F}$ and constraints $\mathbf{G}$ are nice ***algebraic functions*** **of simulation outputs?**

- What (if anything) do we gain from modeling/evaluating $\mathbf{S}$ separately from $\mathbf{F}/\mathbf{G}$?

PARMOO

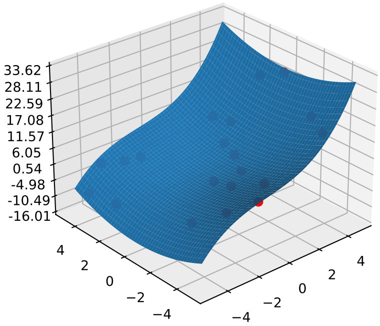
- PARMOO is a Python library for parallel multiobjective simulation optimization
- One of the features of PARMOO is to handle simulations separately from objectives
  - Evaluate sims in $\mathbf{S}$ in parallel
  - Model outputs of $\mathbf{S}$, and compose with $\mathbf{F}$ and $\mathbf{G}$
  - Identify designs $\mathbf{x}$ at which to evaluate sims $\mathbf{S}$

PARMOO continued



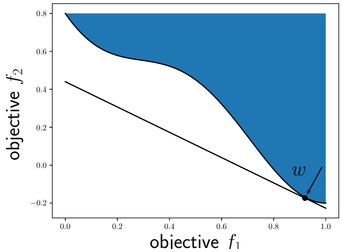
PARMOO Response Surface Model

- Global Search
  - **Latin hypercube search**
  - Global optimization (TBD)
- Global surrogate
  - **Gaussian RBF**
  - Polynomial model (TBD)
  - Triangulation (TBD)
- Optimizers
  - Random search
  - **GPS direct search polling**
  - SGD (TBD)
  - Quasi-Newton (TBD)



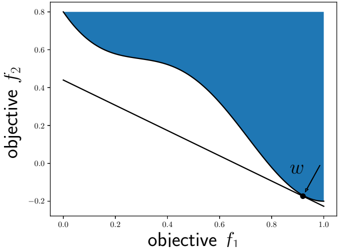
PARMOO Acquisition

- Uniform random weights
- **Fixed weights**

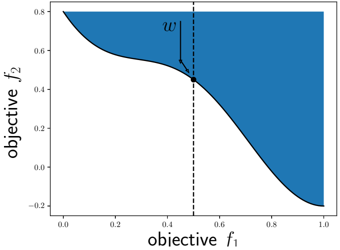


PARMOO Acquisition

- Uniform random weights
- **Fixed weights**



- **Pick random target in the convex hull of nondominated pts, and improve it**
- Other TBD



The Fayans EDF Model

As an example, consider the problem of fitting 9 different observable types for the Fayans EDF model:

- Binding energy;
- Charge radius;
- Diffraction radius;
- Surface thickness;
- Neutron single-level energy;
- Proton single-level energy;
- Differential Radii;
- Neutron pairing gap; and
- Proton pairing gap.

Fitting the Fayans Model

Find parameters $\mathbf{x} \in \mathbb{R}^{13}$ so that the Fayans model $M(\nu;\mathbf{x})$ agrees with data $(\nu_{t,i}, d_{t,i})$:

$$M(\nu_{t,i};\mathbf{x}) \approx d_{t,i} \qquad i = 1,\ldots,n_t; \qquad t = 1,\ldots,9,$$

We have the simulation:

$$S_{t,i}(\mathbf{x}) = M(\nu_{t,i};\mathbf{x}) - d_{t,i} \qquad i = 1,\ldots,n_t; \qquad t = 1,\ldots,9$$

- $m = \sum_{t=1}^{9} n_t = 198$ total sim outputs

R. Bollapragada, M. Menickelly, W. Nazarewicz, J. O'Neal, P.G. Reinhard, and S.M. Wild. Optimization and supervised machine learning methods for fitting numerical physics models without derivatives. Journal of Physics G: Nuclear and Particle Physics 48(2), 2020.

The Fayans Multiobjective Optimization Problem

The problem of fitting the Fayans EDF is formulated as

$$\min_{\mathbf{x}} \sum_{t=1}^{9} \frac{1}{\sigma_t} F_t(\mathbf{x})$$

where

$$F_t(\mathbf{x}) = \sum_{i=1}^{n_t} S_{t,i}(\mathbf{x})^2, \qquad t = 1, \ldots, 9$$

- σ_t is the standard error – errors in measurement and model accuracy
- The correct values of σ_t are unknown

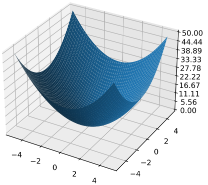
To find σ_t, solve:

$$\min_{\mathbf{x}} (F_1(\mathbf{x}), \ldots, F_9(\mathbf{x}))$$

Sum-of-squares Structure

Problem has a sum-of-squares structure to exploit:

$$F_t = \sum_{i=1}^{n_t} S_{t,i}(\mathbf{x})^2$$



- Use *fully linear models* to approximate $S_{t,i}$ (1st-order approx)
- Pushing $\nabla S_{t,i}(x)$ through F_t gives approximation to the Hessian $\nabla^2 F_t(x)$ † (2nd-order approx)
- Higher-order surrogate approximation $\Rightarrow$ better surrogate optimization, right?

† *H. Zhang, A.R. Conn, and K. Scheinberg. A derivative-free algorithm for least-squares minimization. SIAM Journal on Optimization 20(6), 2010.*

What could go wrong?

Not so fast!

- No single converging sequence $\Rightarrow$ No asymptotic model convergence
- Improved model order could be dominated by a decrease in smoothness

Is modeling $S_{t,i}(\mathbf{x})$ a good idea in practice?
Do we gain anything for the Fayans EDF calibration problem?

3-objective variation

Reduce problem difficulty by bringing down to 3 objectives:

- Binding energy (**sum of 63 squared errors**);
- Std radii (charge radius + diffraction radius, **sum of 80 squared errors**);
- Other (all other quantities of interest, **sum of 55 squared errors**);

The Substitute Problem

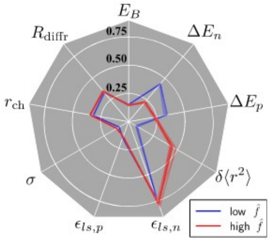
To save compute time, we use a substitute problem

$$\hat{S}_{t,i}(\mathbf{x}) \approx S_{t,i}(\mathbf{x})$$

- Based on 52,079 ε-unique data points (from *Bollapragada et al.*)
- Uses inverse-distance weighting on $k = 14$ $(= n + 1)$ nearest neighbors;
- We are not interested in solutions that are very far away from the minima of $\sum S_{t,i}(\mathbf{x})^2$ found by *Bollapragada et al.*:
  - Sum-of-squared binding energies $(F_1) < 804$;
  - Sum-of-squared std radii $(F_2) < 2090$;
  - Sum-of-other-squares $(F_3) < 613$.

Embedding Fayans EDF in PARMOO

- 2000 pt LHS search
- Gauss RBF surrogates
- 10 total acquisition functions ($\Rightarrow$ 10 sim evals per iteration)
  - 9 Improve random acquisition
  - 1 fixed (equal) weights acquisition function
- GPS polling optimizer
- 10,000 sim budget (-2000 LH search) $\Rightarrow$ 800 iterations



Bollapragada et al. minimized sum-of-squares for all 198 errors, and found 2 solution types (above)

Two Problem Variations

Recall, our goal is to determine whether anything is gained by modeling the individual errors instead of the sum-of-square errors:

Two Problem Variations

Recall, our goal is to determine whether anything is gained by modeling the individual errors instead of the sum-of-square errors:

1. Simulations produce a 198-dimensional error vector
   - $m = 198$ surrogate outputs, 3 objectives $=$ sum-of-squared sim outs
   - Surrogates applied to each error, so sum-of-squares structure is exploited ✓

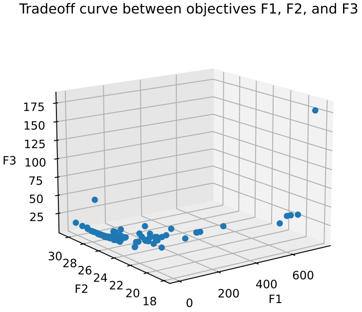
Two Problem Variations

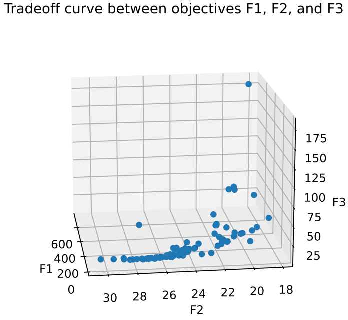
Recall, our goal is to determine whether anything is gained by modeling the individual errors instead of the sum-of-square errors:

1. Simulations produce a 198-dimensional error vector
   - $m = 198$ surrogate outputs, 3 objectives $=$ sum-of-squared sim outs
   - Surrogates applied to each error, so sum-of-squares structure is exploited ✓
2. Simulations given as 3-dimensional sum-of-squared error objective vector
   - Objectives are given as the identity function on each simulation output
   - Surrogates applied to true objectives, so sum-of-squares structure is not exploited ✗

Full Solution

Full solution set: Nondominated objective values from original 52,079 data points + 100,000 new evaluations

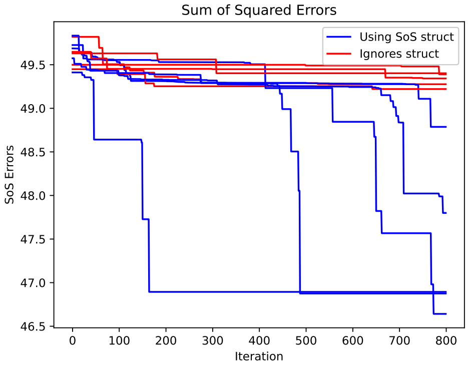




Performance Measures

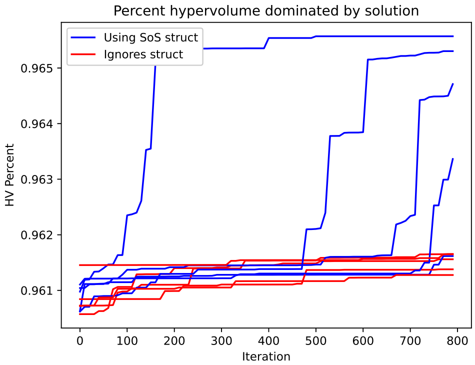
Minimum sum-of-squares (measures convergence of one fixed AcquisitionFunction)

- Note: This is the only metric expected to reach asymptotic convergence rate



Performance Measures

Hypervolume indicator (proportion of hypercube between upper bounds and the origin that is dominated by each solution set)



Conclusions

- The sum-of-squares structures is successfully exploited in this problem by modeling simulations separately from objectives/constraints
- Effective for both accelerating convergence to a single Pareto point and for finding "isolated" Pareto points

Conclusions

- The sum-of-squares structures is successfully exploited in this problem by modeling simulations separately from objectives/constraints
- Effective for both accelerating convergence to a single Pareto point and for finding "isolated" Pareto points

Future work

- Investigate the full 9 objective problem
- Add an acquisition function option for setting adaptive targets on the Pareto front
- Continue to improve PARMOO/add other surrogates and better optimizers
- Investigate other structures (such as heterogeneous objectives)

Questions

Introduction and Motivation
- Notation
- The Simulation Function

PARMOO
- Python Structure
- Algorithmic Components

The Fayans EDF Problem
- Problem Description
- Theoretical Benefits

Experiments
- Reduced Difficulty Problem
- Embedding Fayans EDF in PARMOO
- Results and Conclusions

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, SciDAC program under contract number DE-AC02-06CH11357.