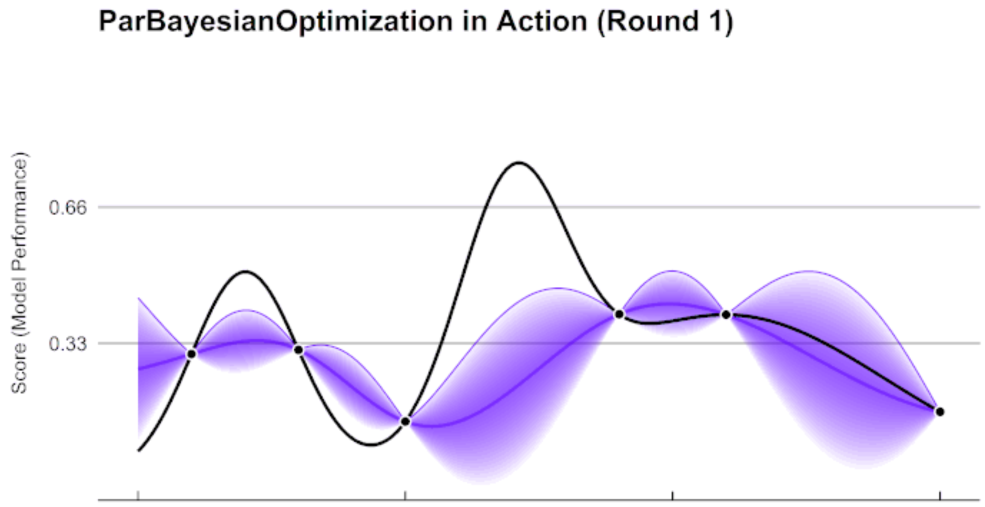
Bayesian Optimization / Machine Learning for GeoPolymer Formulation

Dustin Lobner
Senior Staff Engineer – Innovation
Advanced Development – Materials
Kohler Co – Kitchen and Bath
Kohler, WI USA

Outline

- Introduction
  - What is Bayesian Optimization?
  - The sample problem to be solved – particle size distribution
- Methodologies
  - Ladder testing
  - DOEs
  - Machine Learning – Bayesian Optimization
- Goal: convince you to look into using this method



Source – Github - ParBayesianOptimization

What is Bayesian Optimization?

Bayesian optimization is a machine learning method for finding the **best input settings of an expensive-to-evaluate function** when each experiment, simulation, or measurement costs time or money.

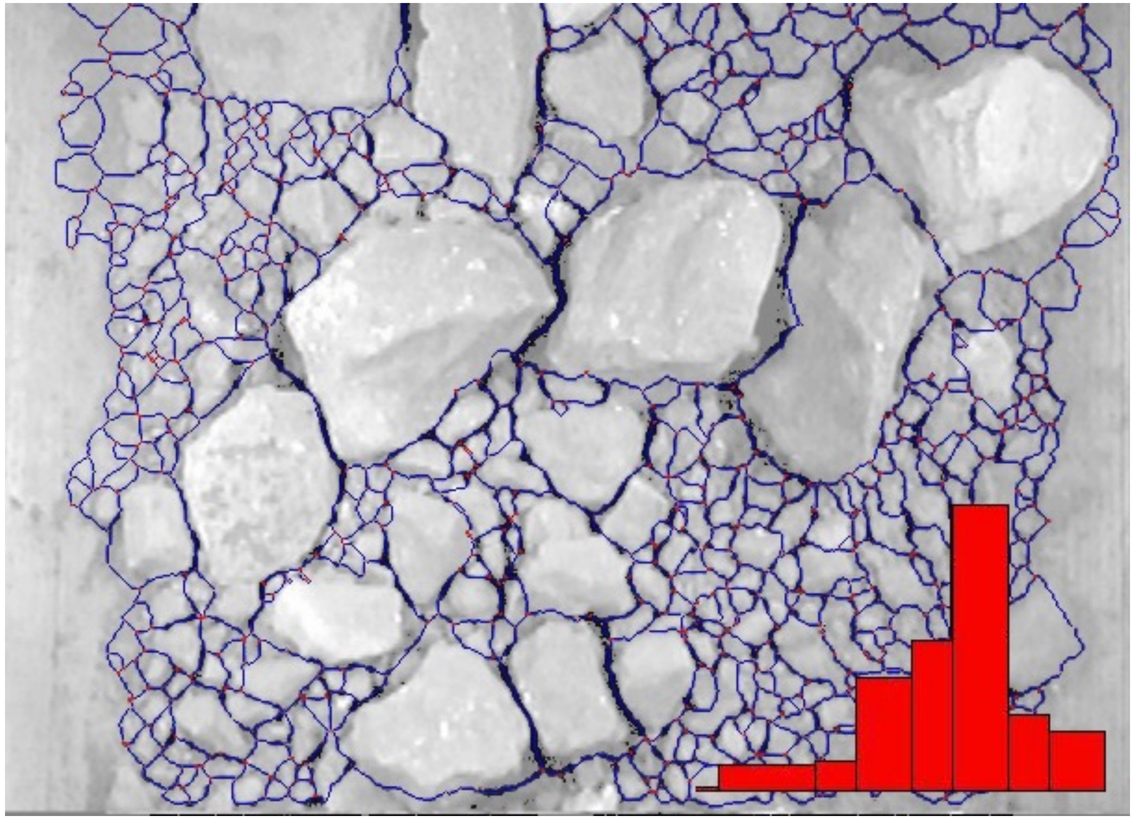
“Given what I know, what is the next best experiment to run?”

It's commonly used for:

- Hyperparameter tuning in machine learning
- <u>Engineering design optimization</u>
- Drug discovery
- A/B testing and experimentation
- Tuning manufacturing processes

Introduction – Example problem to be solved

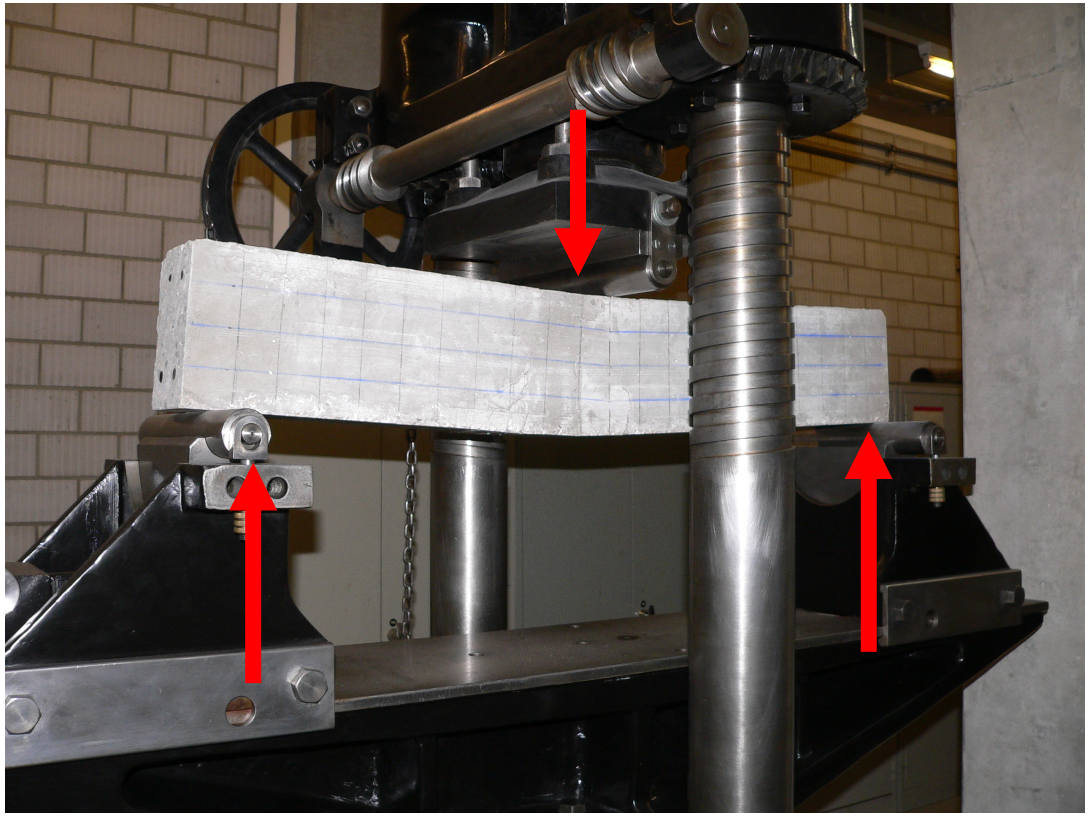
- Particle Size Distribution of aggregate fill
  - Critical to many properties of GeoPolymers
    - Maximize Strength
    - Minimize Shrinkage and Warpage
    - Maximize Processability
- Plus:
  - Admixes
  - Pigmenting
  - Processing
- Potentially >10 inputs



Source: Wikipedia.com

Introduction – Problem to be Solved (cont.)

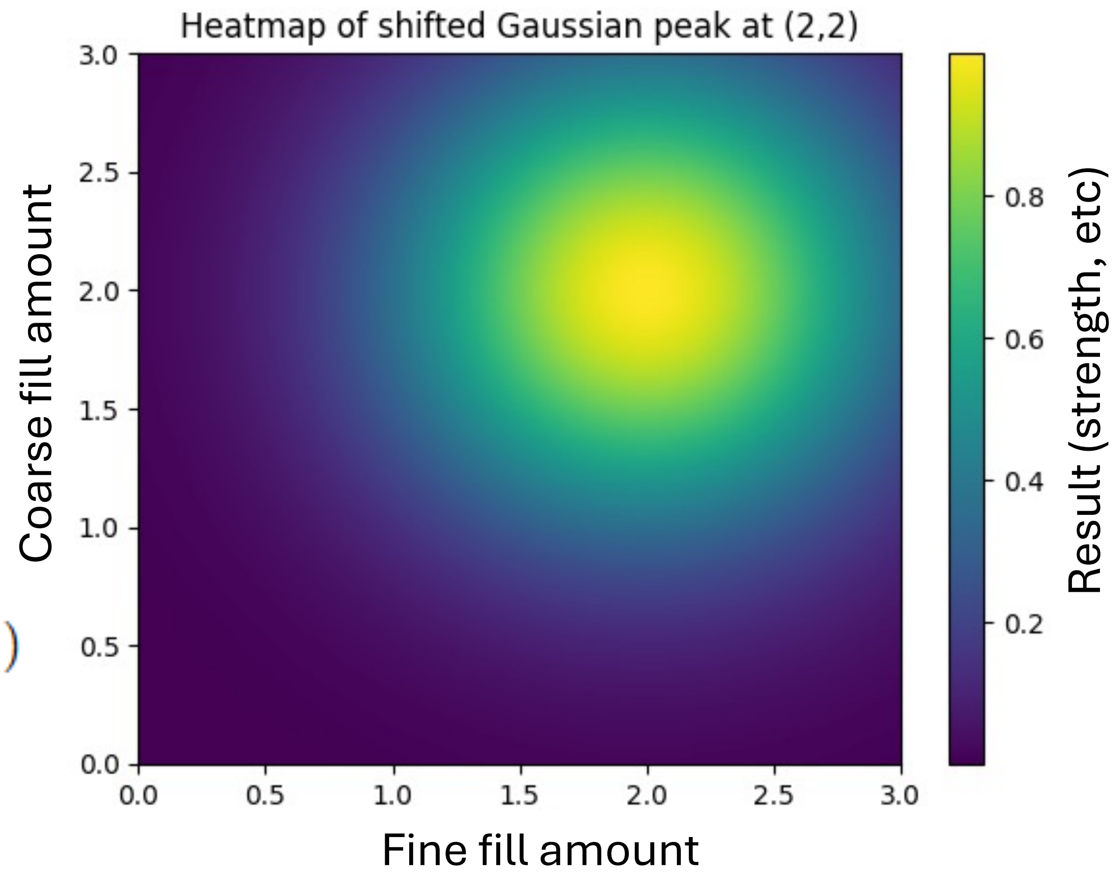
- Outputs
  - Strength (3-point bend)
  - Impact energy absorption (Charpy)
  - Processability (Flow)
  - Cost
- **How do we take all these inputs and trial them in an efficient manner to get the best results?**



3-point bend (source: Wikipedia.com)

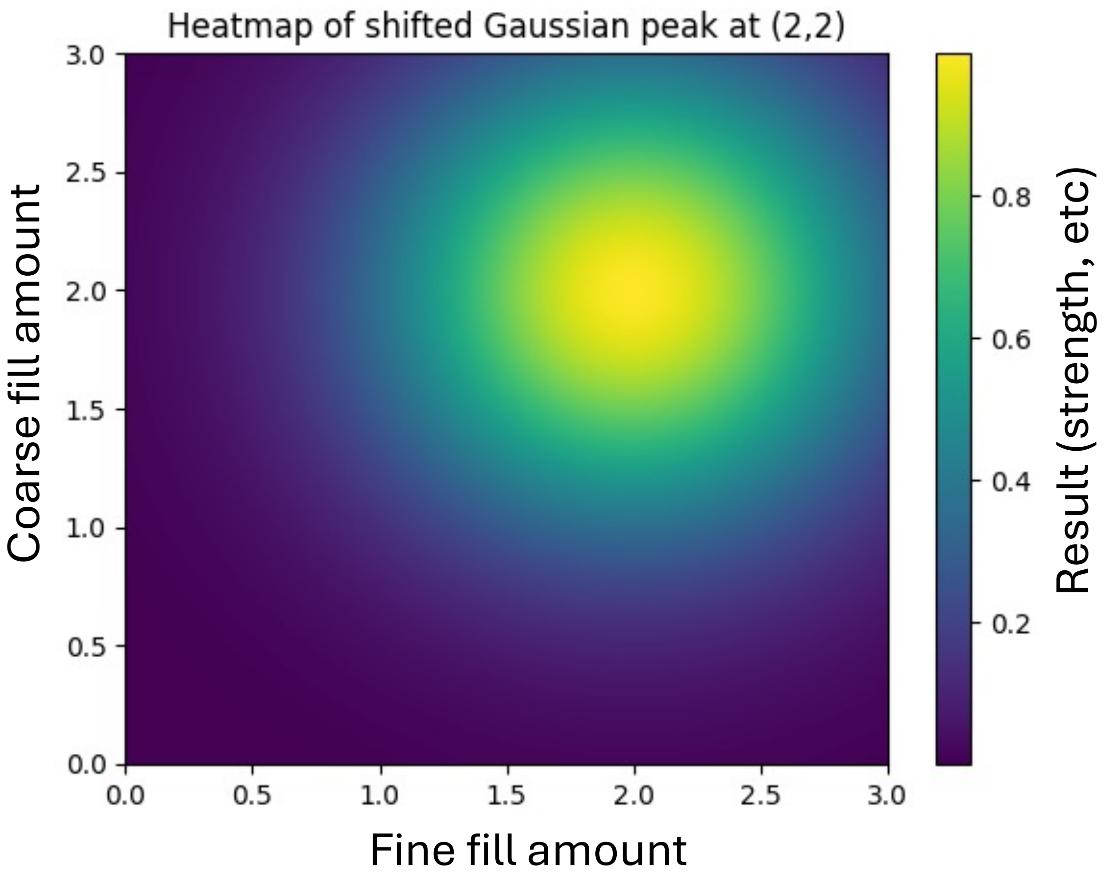
Introduction – Made Up Data for Discussion

- 2 Inputs (two particle sizes)
  - Y-axis: "Coarse Fill"
  - X-axis: "Fine Fill"
  - Range: 0 to 3 for each
- 1 Output Z
  - Z-axis / Color: 0 to 1
  - maximal at (2,2)
- Heatmap equation $z = e^{-((x-2)^2+(y-2)^2)}$
- Use to discuss difference analysis techniques



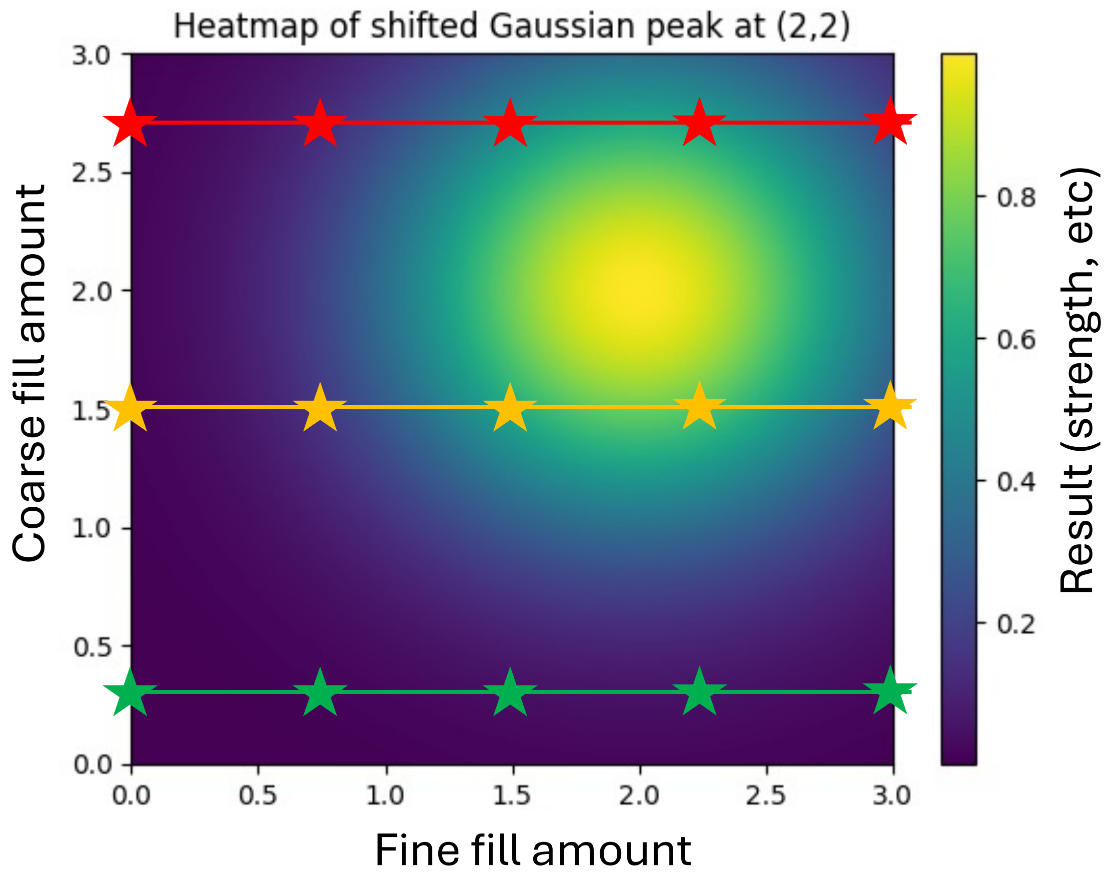
Methodologies Introduction

- Goal – maximize the function - find the maximum at (2,2)
- 3 analysis techniques
  - Ladder Studies
    - "Only change one thing at a time!"
  - DOEs (Design of Experiment)
    - Many techniques here to screen variables, optimize in a space, cut sample counts, etc)
  - Machine Learning (Bayesian Optimization)
    - Newer field and topic of discussion



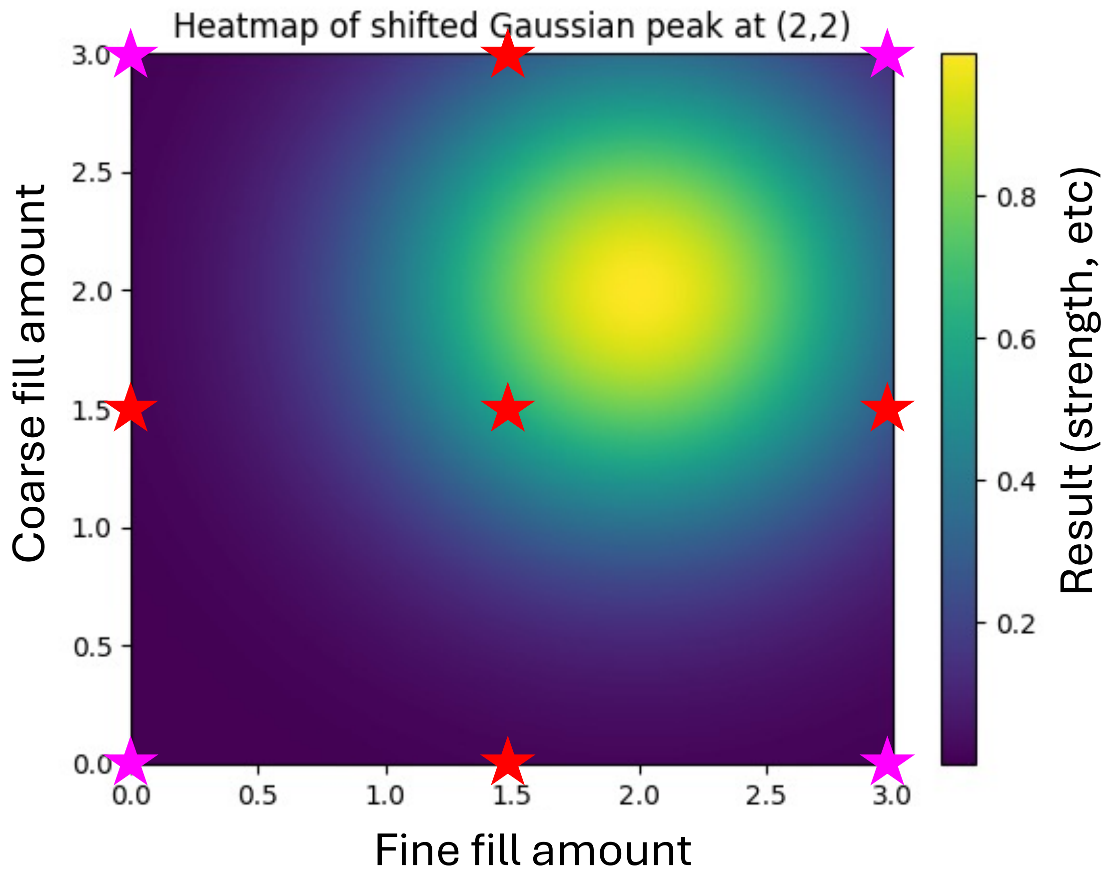
Methodology – Ladder Studies

- Basic Principle – Only change one thing at a time
- Pros –
  - Easy to understand what is going on
  - Simple to run – mix a batch, cast a part, add more of the component, cast a part, etc
- Cons –
  - Only 1 variable changing at once – slow progress
  - Misses interactions between variables
  - Incredibly slow for more than a couple of inputs
- This example – 15 trials, have gotten to 50% of max
- <u>**How do you do this with >10 variables?**</u>



Methodology – DOE: Design of Experiment

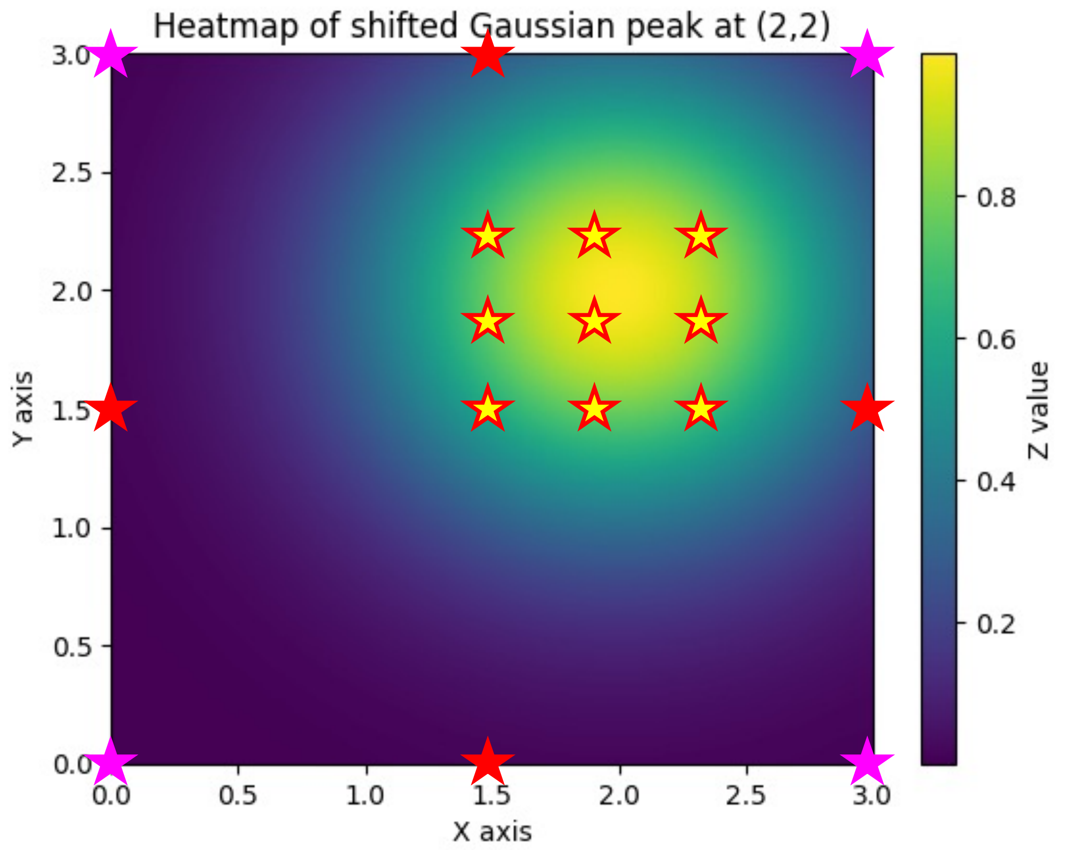
- Basic Principle – Organized way of setting up tests to maximize knowledge gain from work done
- Pros –
  - Organized method of bringing order from chaos
  - Well known technique, software available to help
  - Can be "tuned" to reduce the number of samples needed
- Cons –
  - Don't get significant direction until all the data is available



Methodology – DOE: Design of Experiment

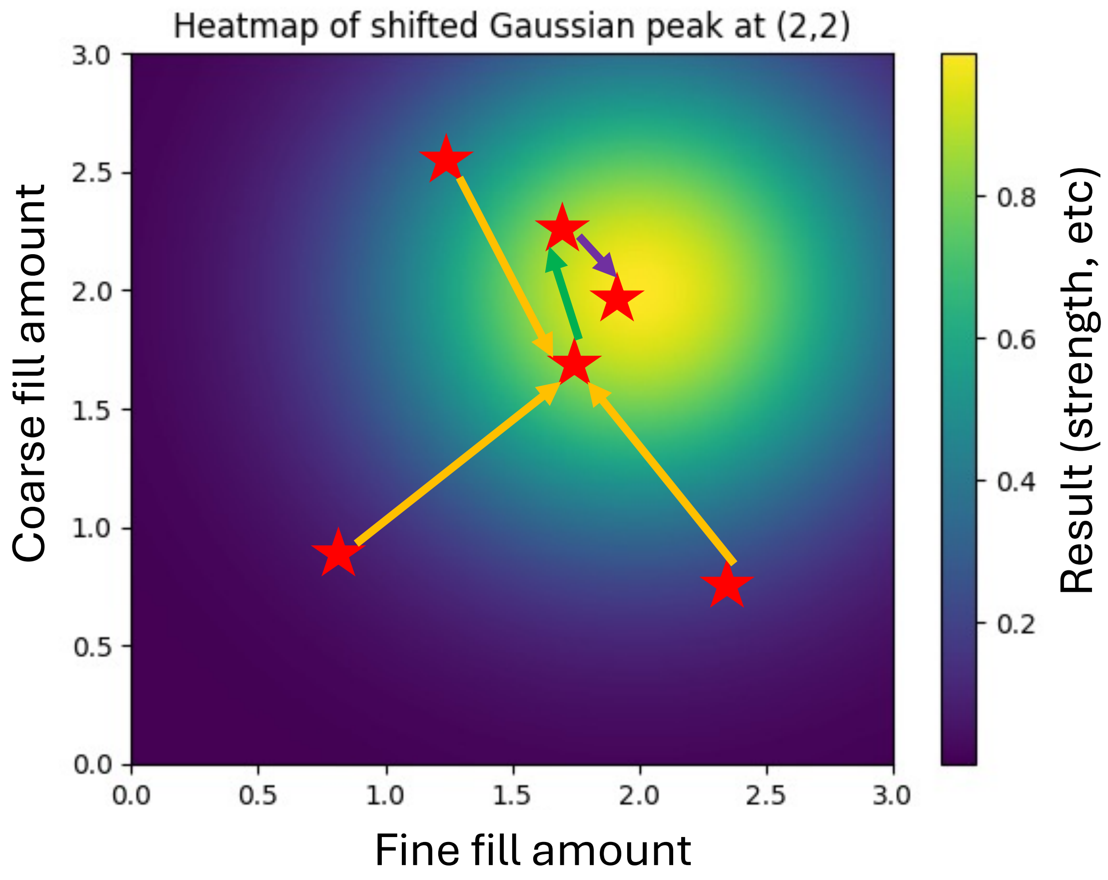
- This example
  - 18 trials (see additional 9 at right)
  - Optimized after two 3^2 test programs

- How do you do this in the real world with particle size distribution?
  - 3 level full factorial, 10 inputs – 3^{10} samples
    - 59,049 samples
  - All significant, all dependent on each other
- <u>Can't screen out variables – they all have effects and they all interact</u>



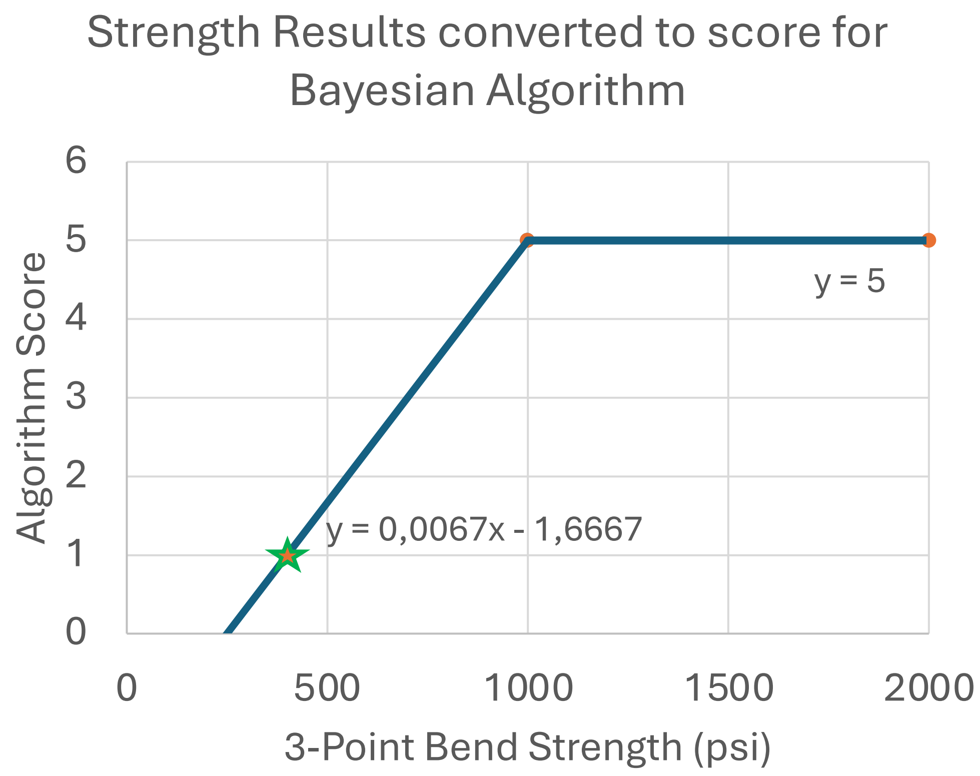
Methodology – Bayesian Optimization

- Basic Principle – Machine learning algorithms know more about finding patterns in data than you ever will
  - Let the algorithm guide the work
- Pros –
  - After an initial few samples, you are able to do 1 test at a time for optimum speed of convergence
  - On average, far better speed (efficiency) and quality of results than DOEs and ladder studies
  - **You have to understand what is "good" and "bad" in advance**
- Cons –
  - Lose some interpretability
  - **You have to understand what is "good" and "bad" in advance**



Methodology – Bayesian Optimization Grading

- Algorithm only takes in one result, a number
  - In free software I'll link to, that is 0-5 for "zero to five stars", when originally used in a brownie recipe
- Need to map your experimental results to 0-5 – the algorithm will then try to get the output equal to 5
- Example at right – 3-point bend failure stress, trying to replace a material
  - Current material has 400 psi bend strength – set this equal to 1
  - "Great" would be 1,000 psi – set this equal to 5
  - Anything over 1,000 psi is still 5
    - Don't want to swamp other results



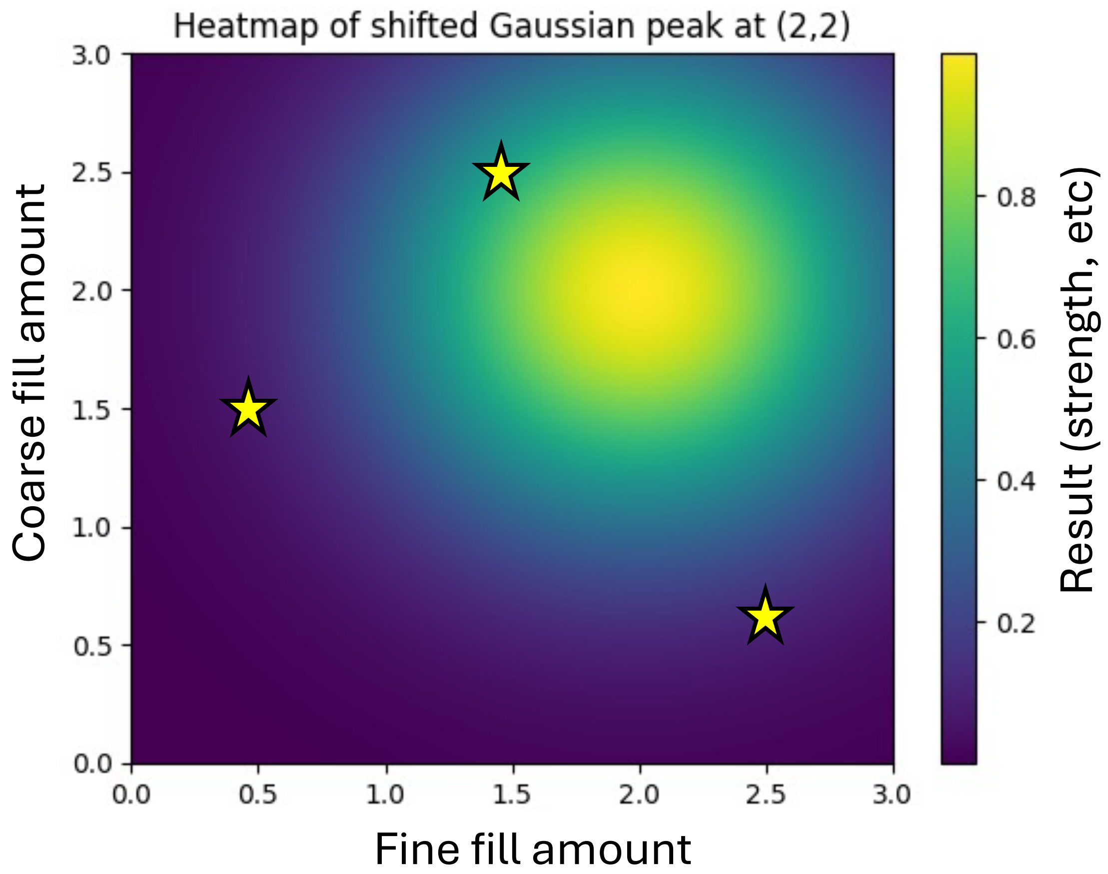
Methodology – Bayesian Optimization Grading

- What if we have more than one output?
  - Examples – processability score, impact strength, compression strength, cost of material..
- Map other results to their own 0-5 scoring system
  - Can be subjective – “how easy was this material to process, on a 0-5 scale?”
- Combine results
  - Straight average (easiest)
  - Weighted average (do you care more about one topic than another?)
- Feed this single number back into the algorithm as the result for that formulation

You must put in the effort to understand what is “good” and what is “bad” before you start experimenting, and you must model this mathematically and understand why you model it this way

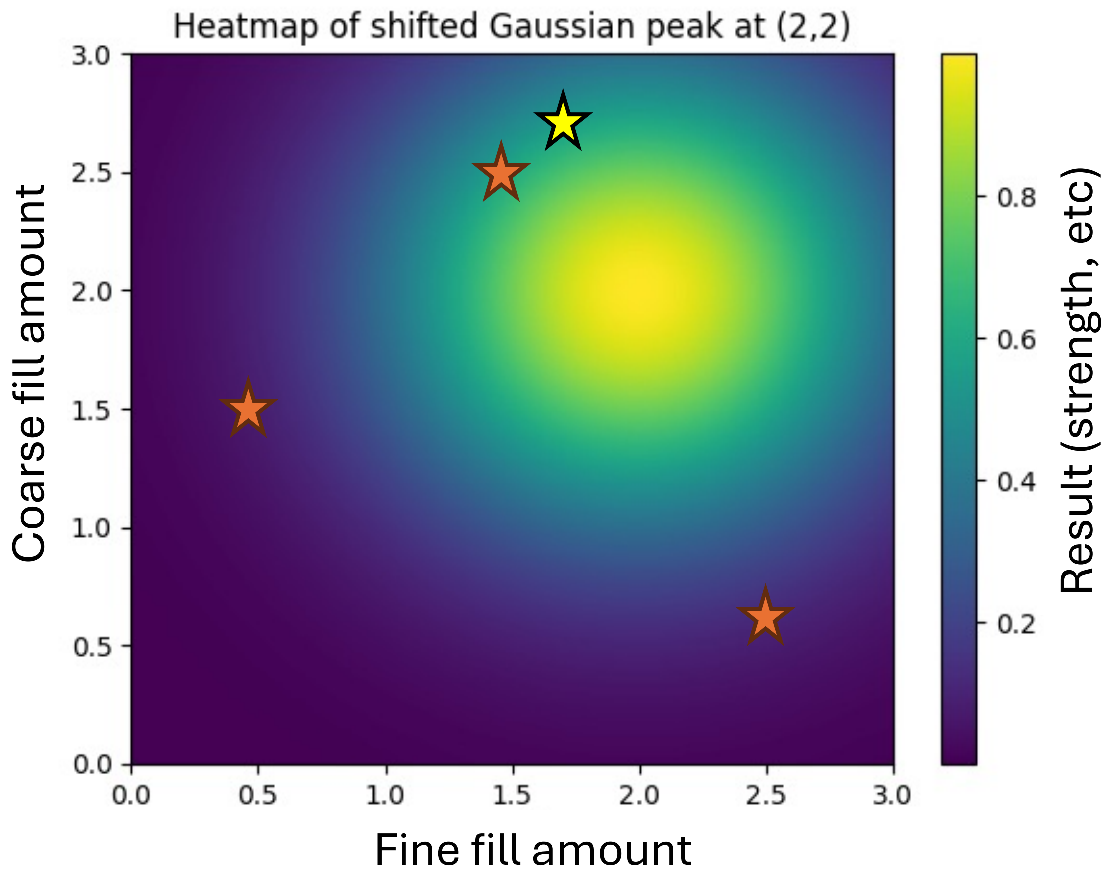
Methodology – Bayesian Optimization Example

- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410



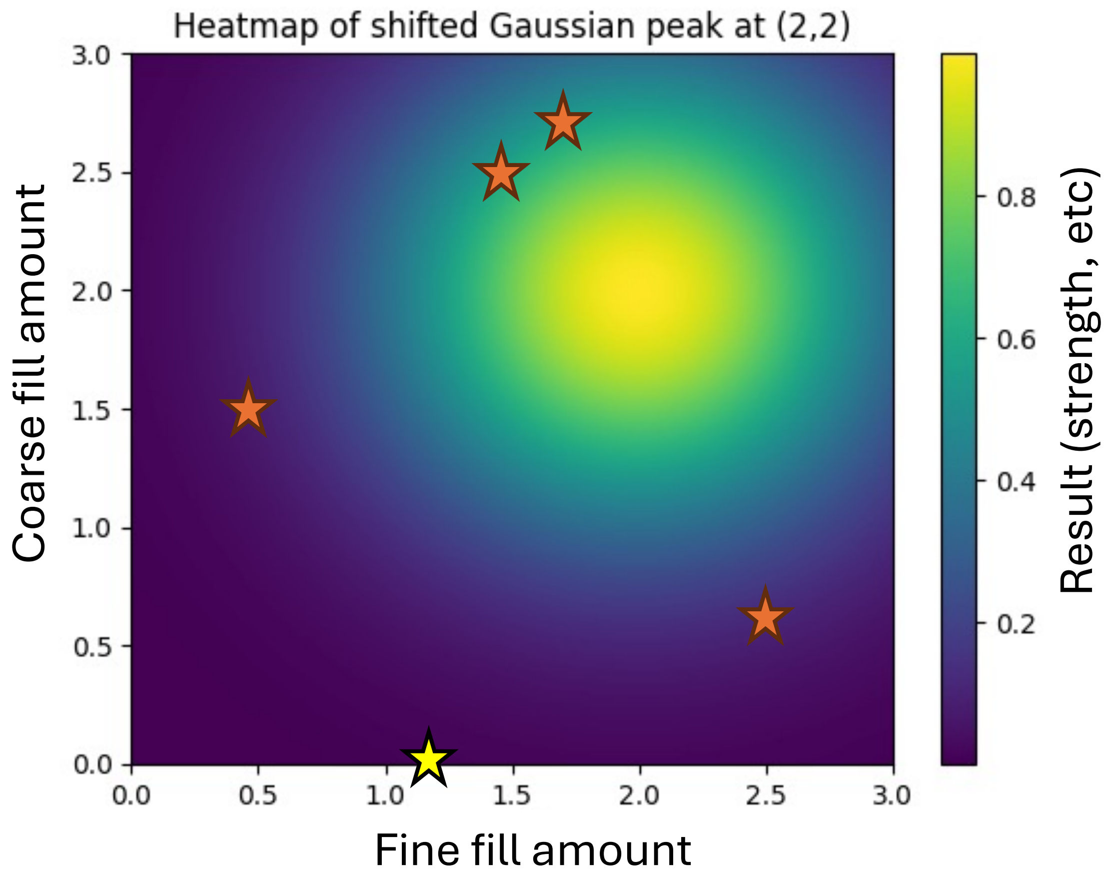
Methodology – Bayesian Optimization Example

- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410
- Iterations
  - (1.687, 2.773) = .499 * 5 = 2.494



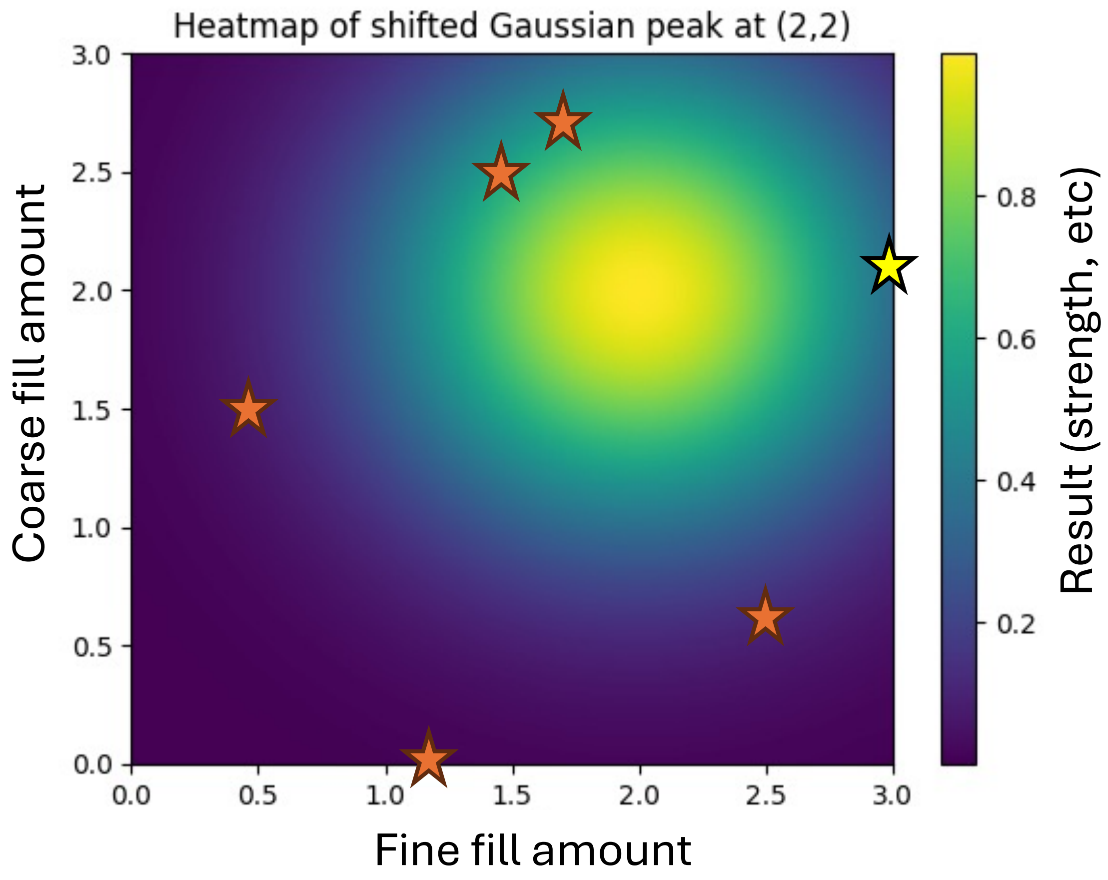
Methodology – Bayesian Optimization Example

- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410
- Iterations
  - (1.687, 2.773) = .499 * 5 = 2.494
  - (1.296, 0) = 0.011 * 5 = 0.056



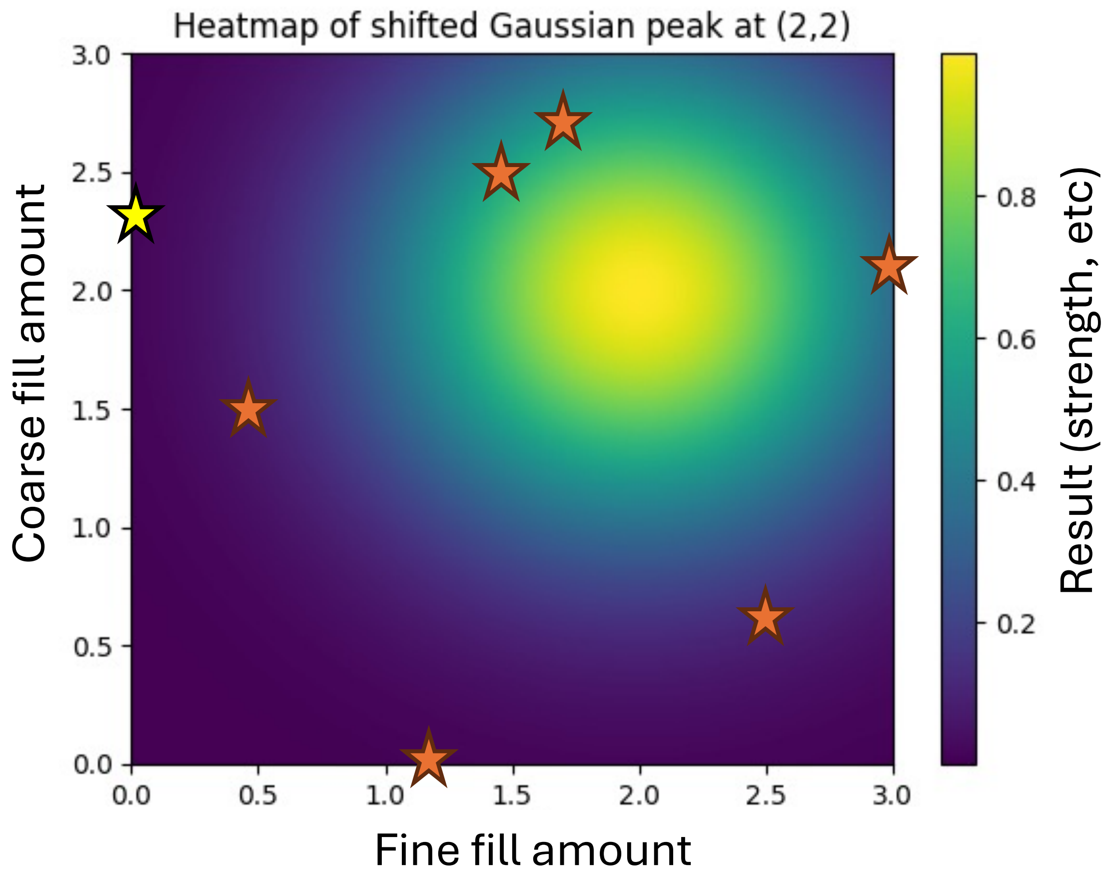
Methodology – Bayesian Optimization Example

- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410
- Iterations
  - (1.687, 2.773) = .499 * 5 = 2.494
  - (1.296, 0) = 0.011 * 5 = 0.056
  - (3, 2.166) = 0.357 * 5 = 1.789



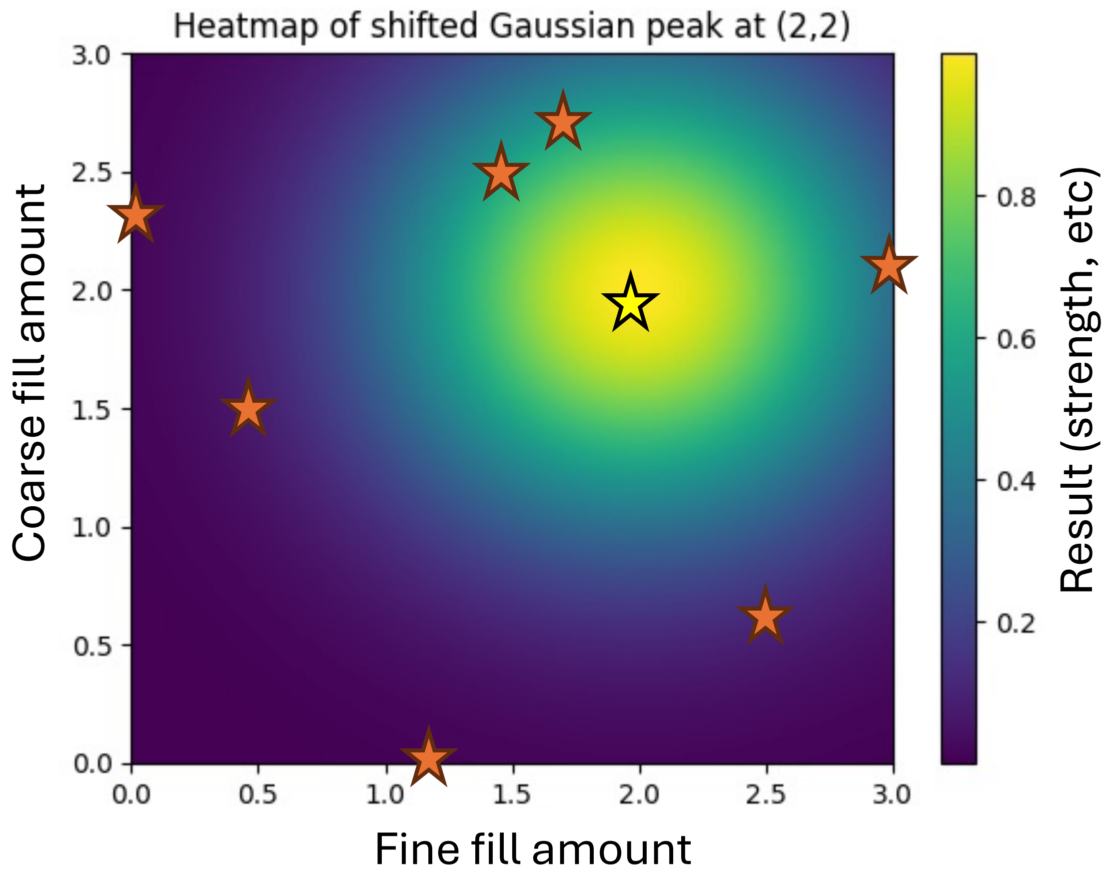
Methodology – Bayesian Optimization Example

- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410
- Iterations
  - (1.687, 2.773) = .499 * 5 = 2.494
  - (1.296, 0) = 0.011 * 5 = 0.056
  - (3, 2.166) = 0.357 * 5 = 1.789
  - (0, 2.317) = 0.0166 *5 = 0.082



Methodology – Bayesian Optimization Example

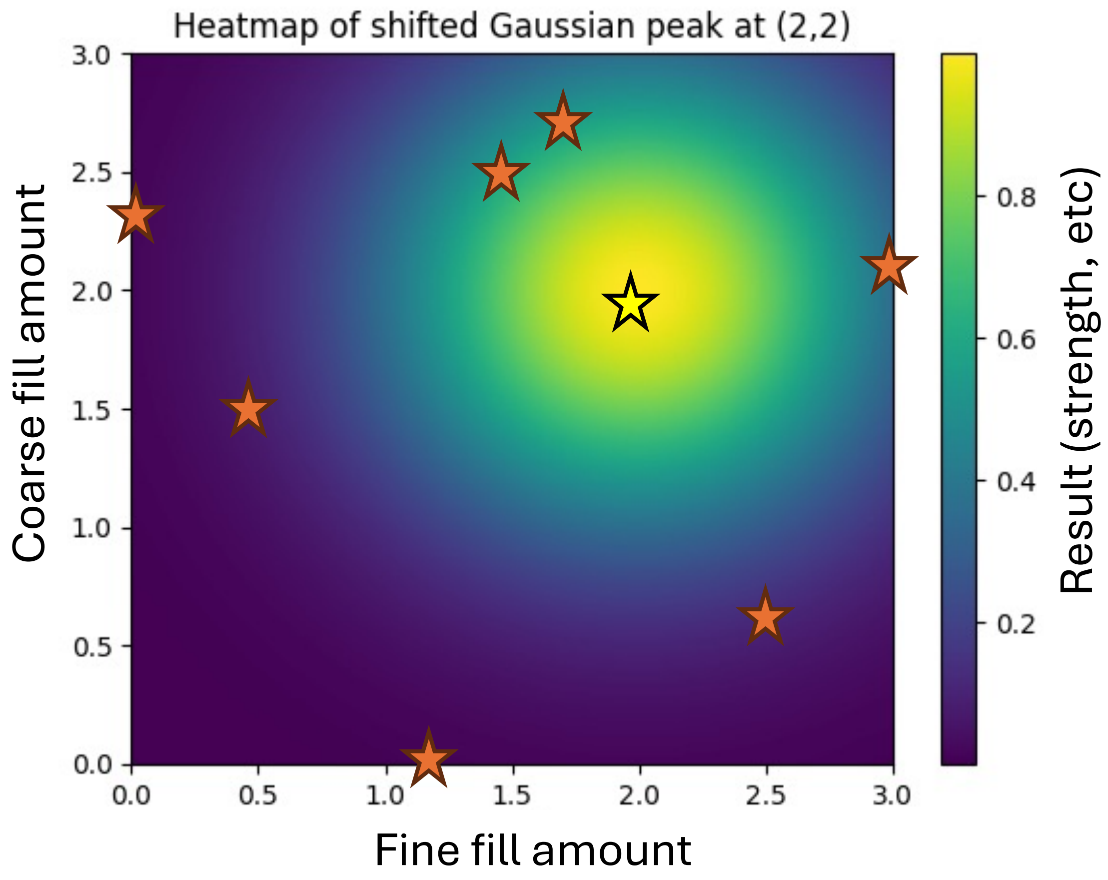
- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410
- Iterations
  - (1.687, 2.773) = .499 * 5 = 2.494
  - (1.296, 0) = 0.011 * 5 = 0.056
  - (3, 2.166) = 0.357 * 5 = 1.789
  - (0, 2.317) = 0.0166 *5 = 0.082
  - (1.896, 1.935) = 0.985 * 5 = 4.925



Methodology – Bayesian Optimization Example

- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410
- Iterations
  - (1.687, 2.773) = .499 * 5 = 2.494
  - (1.296, 0) = 0.011 * 5 = 0.056
  - (3, 2.166) = 0.357 * 5 = 1.789
  - (0, 2.317) = 0.0166 *5 = 0.082
  - (1.896, 1.935) = 0.985 * 5 = 4.925

<u>8 total experiments to get within 1.5% of max</u>

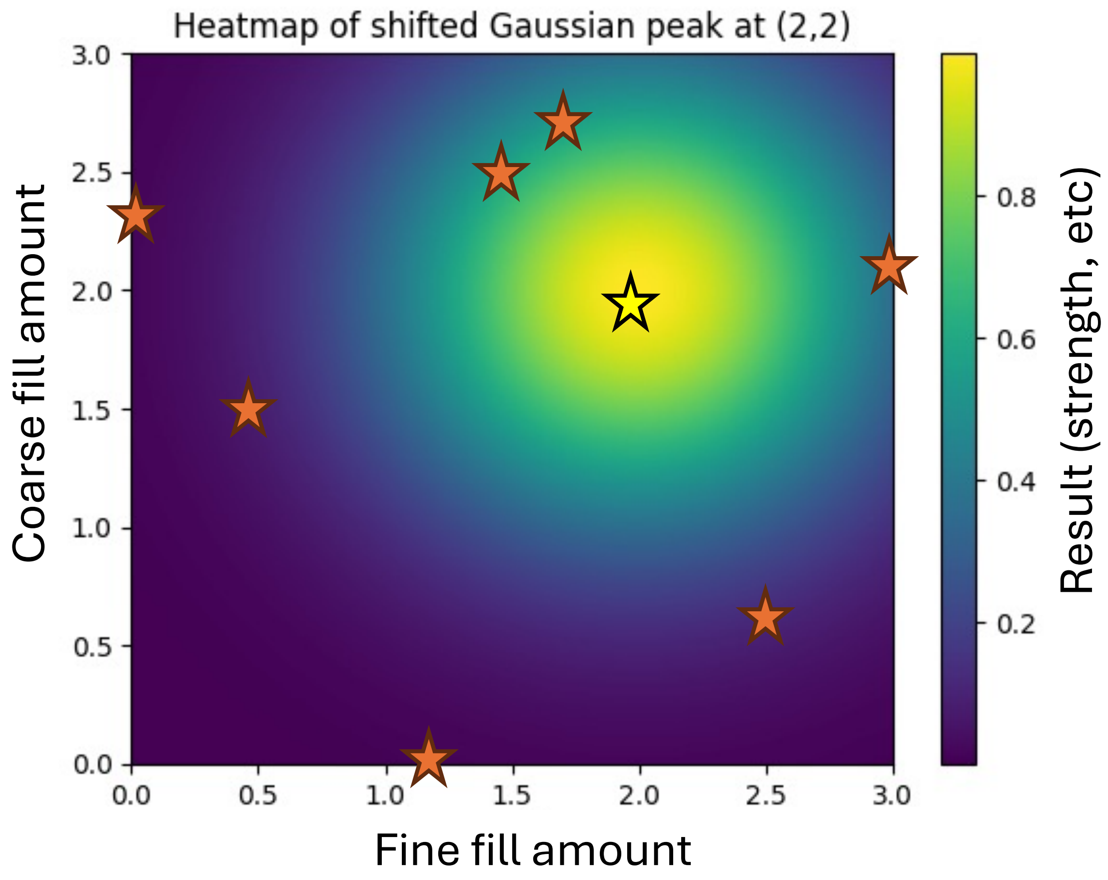


Methodology – Bayesian Optimization Example

- 3 starting points (fine, coarse)
  - (2.5, 0.5) = 0.082 * 5 = 0.410
  - (1.5, 2.5) = 0.606 * 5 = 3.033
  - (0.5, 1.5) = 0.082 *5 = 0.410
- Iterations
  - (1.687, 2.773) = .499 * 5 = 2.494
  - (1.296, 0) = 0.011 * 5 = 0.056
  - (3, 2.166) = 0.357 * 5 = 1.789
  - (0, 2.317) = 0.0166 *5 = 0.082
  - (1.896, 1.935) = 0.985 * 5 = 4.925

<u>8 total experiments to get within 1.5% of max</u>

- 15 to get close on a ladder, 18 for DOE

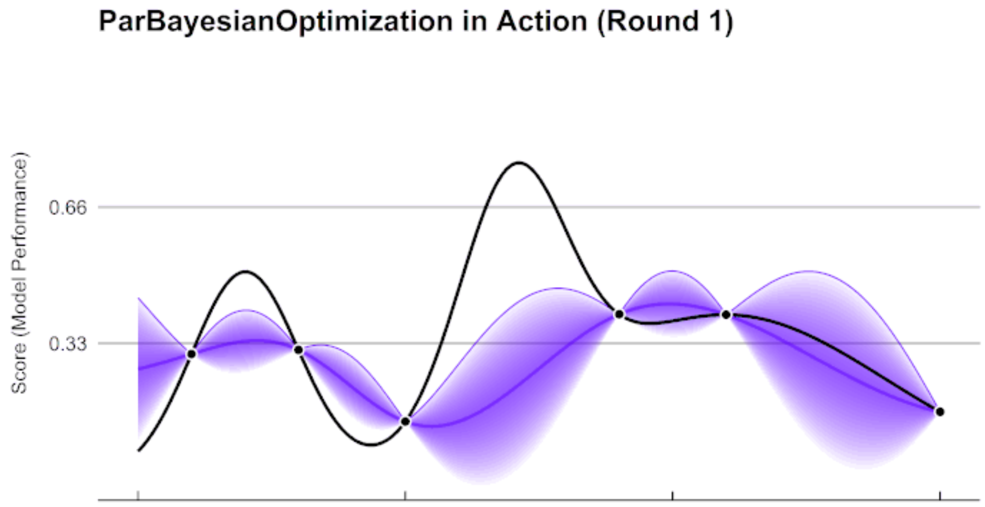


Example Successes

- Outside of Kohler
  - 3D printed structure – DOE "optimized" in 1800 trials, BO tied DOE in 32 runs and beat DOE in 64 runs
  - Chemical reaction – BO found reaction pathways that "shouldn't exist"…but did
- Within Kohler
  - Material formulation – 9 inputs – DOEs: 19K tests BO: 13 to optimum
  - CFD – valve flow optimized for different pressures – reduce SKU counts by 1/3 across multiple product lines
  - CFD – showed that design idea wasn't possible

Conversations around Bayesian Optimization

- You DO work on multiple variables at once – the algorithm is smarter than you are
- You lose the ability to easily talk about the statistical power – no <u>uniform</u> confidence intervals, etc
- Typically used on higher dimensional problems – hard to visualize
- More parameters to adjust
  - Most impactful is alpha (α) – are we exploring or exploiting?



Source – Github - ParBayesianOptimization

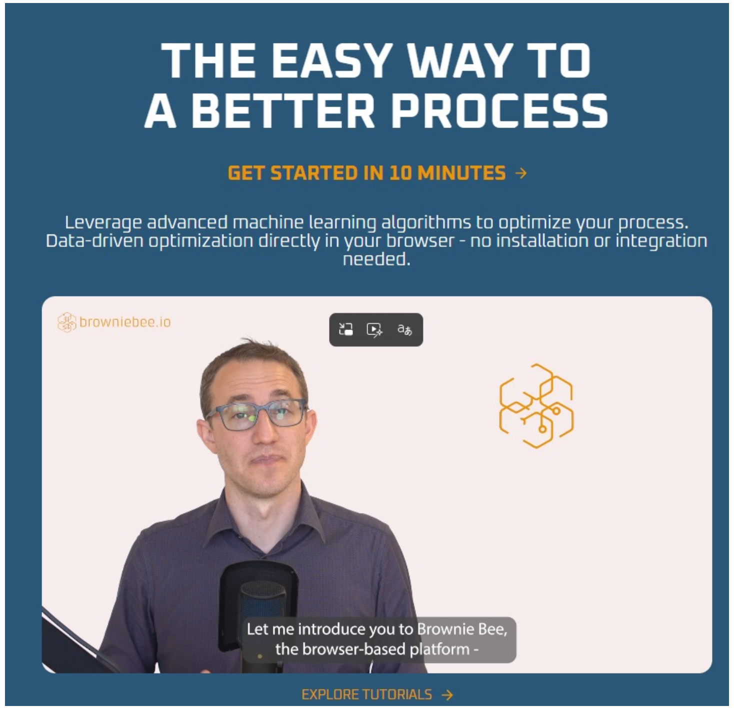
Getting Started in Bayesian Optimization

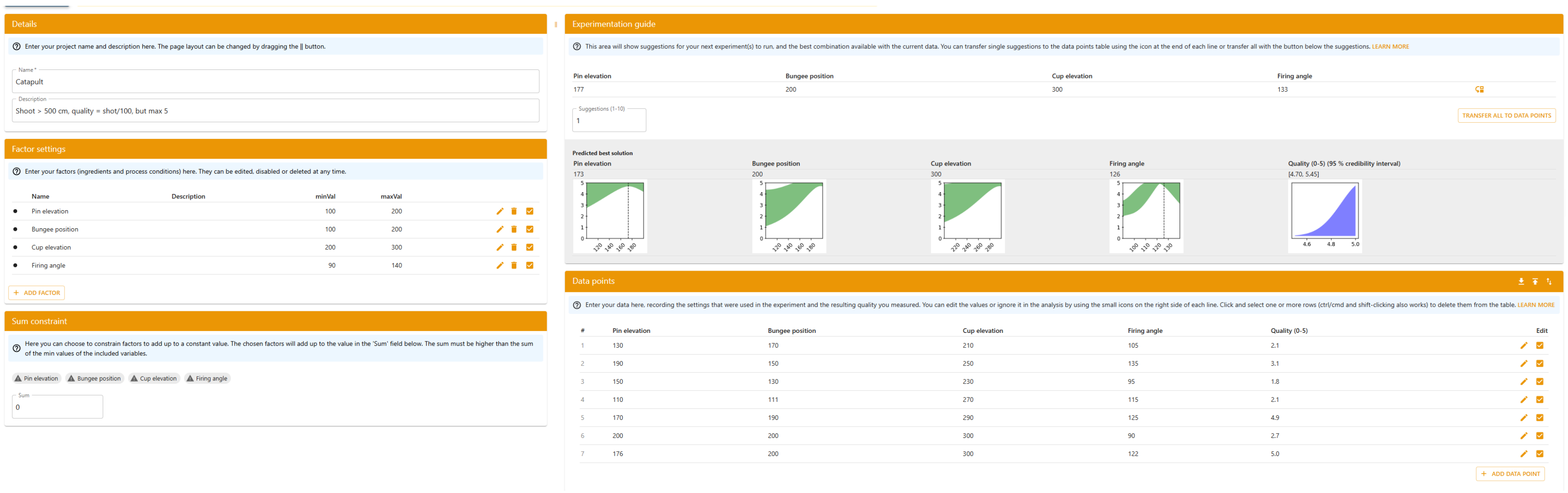
- For **non-Python** Users, Danish Technological Institute website: **Browniebee.io**
  - 0-5 stars scale
  - Free to try
  - €69/month (allows saving data)
  - Example problems to demonstrate
  - Powerful, but many of the features/decisions are made for you – easy to get started, limiting for deeper analysis

- For **Python** Users – multiple Bayesian Optimization packages on GitHub
  - BayesianOptimization
  - Obsidian
  - BoTorch (in PyTorch)
  - GPFlowOpt (in GPFlow)
  - Bofire
  - BOXVIA (has a GUI)
  - ParBayesian Optimization

  - Whatever scale you want!

Getting Started: BrownieBee.io





Literature: Bayesian Optimization Examples

Thank you!

Contact information:

Dustin Lobner
Dustin.Lobner@Kohler.com
Mobile: +1-920-226-5967 (contact via WhatsApp)