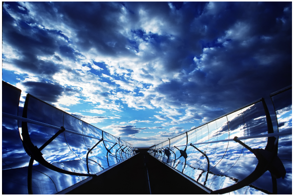
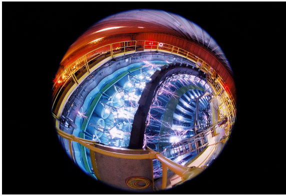
Exceptional service in the national interest

Sandia National Laboratories





Versatile Gaussian process and Bayesian optimization for multiscale computational materials science applications

Mathematics in Computation (MiC) Seminar. March 24, 2022, Oak Ridge, TN.

Anh Tran (Sandia National Laboratories)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.
SAND NO. 2022-XXXXXX

boratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LL
Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE

Acknowledgment

Joint work with

- SNL: Tim Wildey, Mike Eldred, Bart G van Bloemen Waanders, Kathryn Maupin, John Mitchell, Laura Swiler, Julien Tranchida, Aidan Thompson, Theron Rodgers, Hojun Lim, David Montes de Oca Zapiain
- ORNL: Hoang Tran
- Georgia Tech: Yan Wang, Stefano Travaglino, Wei Sun
- Others: Scott McCann (Xilinx), John Furlan (GIW), Krishnan Pagalthivarthi (GIW), Robert Visintainer (GIW)

Funded by

- NSF
- DOE/Office of Science/ASCR
- Sandia ASC and LDRD Program

The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Bayesian optimization (animation)

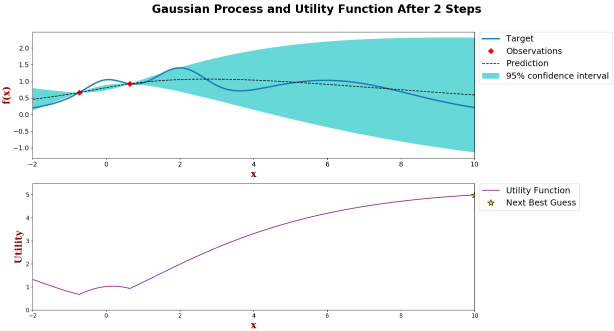


Figure 1: Bayesian optimization - Iteration 1

Bayesian optimization (animation)

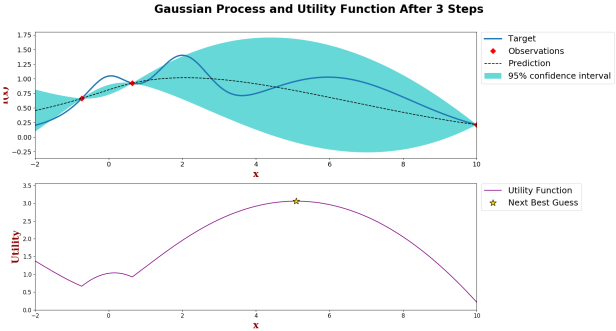


Figure 2: Bayesian optimization - Iteration 2

Bayesian optimization (animation)

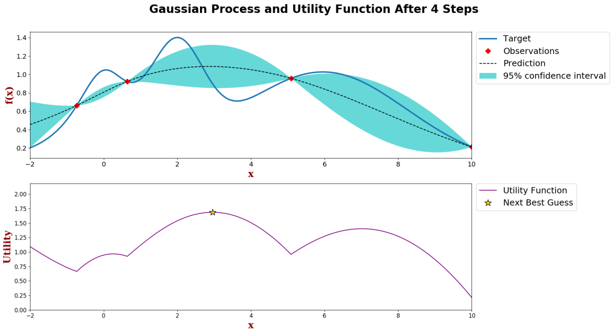


Figure 3: Bayesian optimization - Iteration 3

Bayesian optimization (animation)

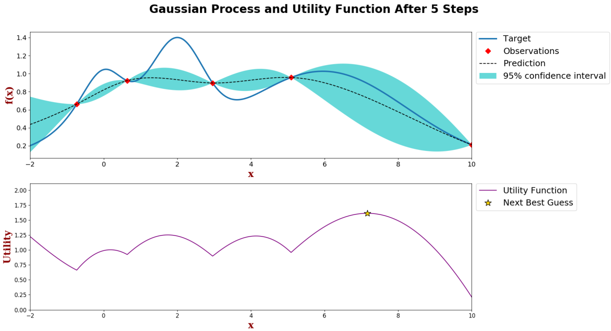


Figure 4: Bayesian optimization - Iteration 4

Bayesian optimization (animation)

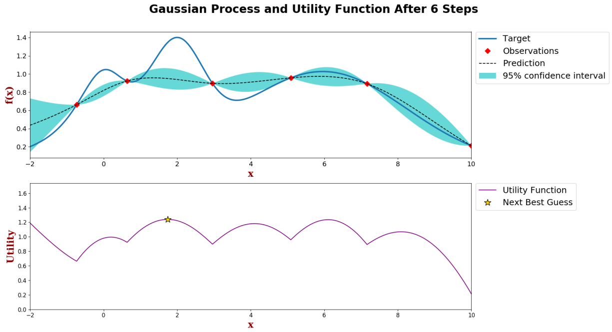


Figure 5: Bayesian optimization - Iteration 5

Bayesian optimization (animation)

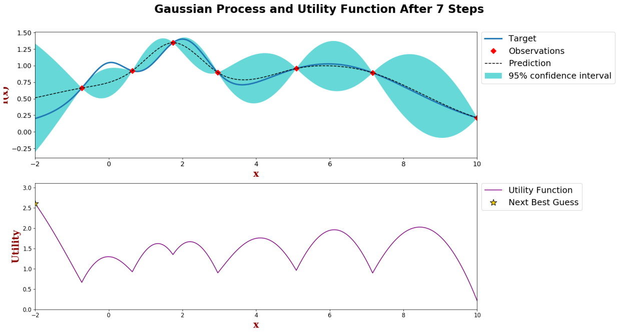


Figure 6: Bayesian optimization - Iteration 6

Bayesian optimization (animation)

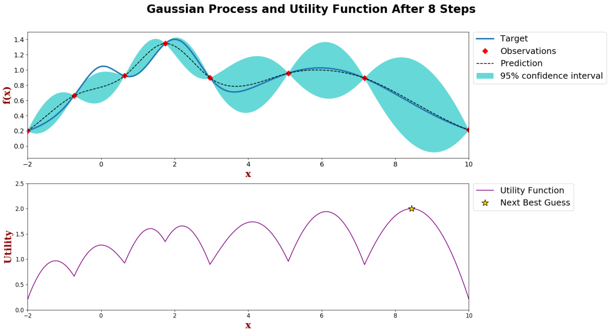


Figure 7: Bayesian optimization - Iteration 7

Bayesian optimization (animation)

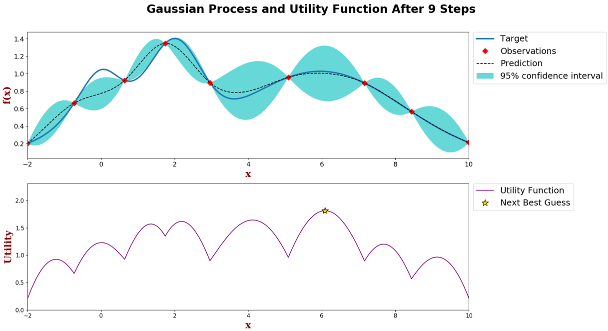


Figure 8: Bayesian optimization - Iteration 8

Bayesian optimization (animation)

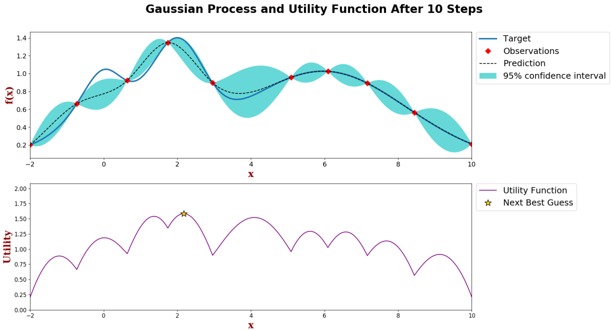


Figure 9: Bayesian optimization - Iteration 9

Bayesian optimization (animation)

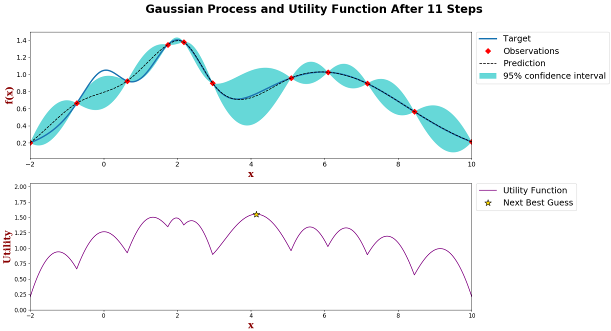


Figure 10: Bayesian optimization - Iteration 11

Bayesian optimization (animation)

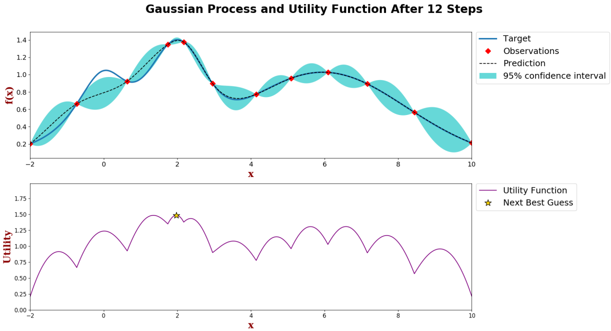


Figure 11: Bayesian optimization - Iteration 11

Bayesian optimization (animation)

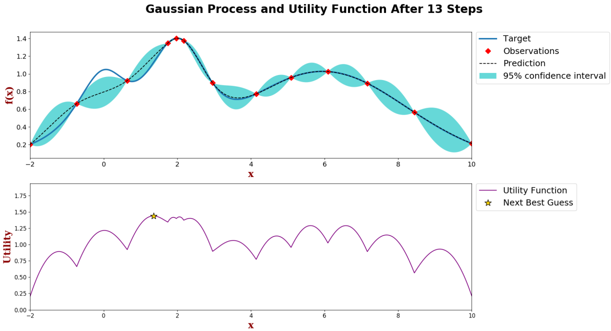


Figure 12: Bayesian optimization - Iteration 12

Bayesian optimization (animation)

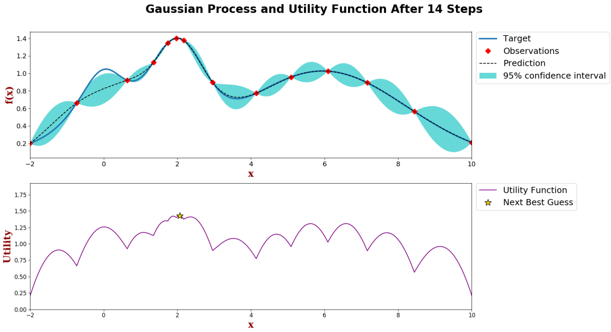


Figure 13: Bayesian optimization - Iteration 13

Bayesian optimization (animation)

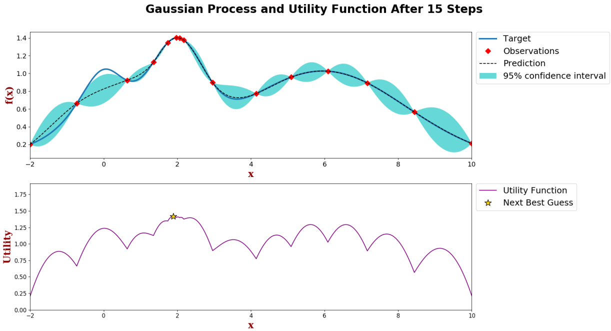


Figure 14: Bayesian optimization - Iteration 14

Bayesian optimization (animation)

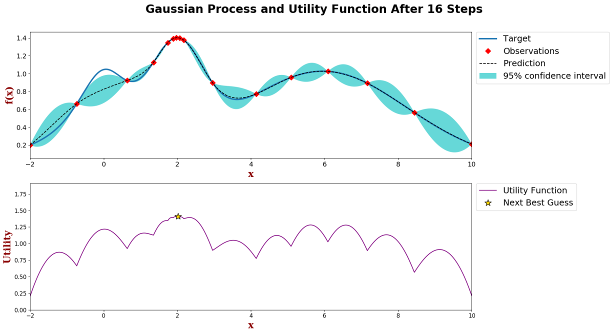


Figure 15: Bayesian optimization - Iteration 15

Advantages/Disadvantages

Bayesian optimization in a nutshell

Bayesian optimization = Gaussian process + sampling strategy

Advantages:

- optimize with uncertainty consideration (e.g. noisy observations)
- active machine learning (balance exploration-exploitation)
- derivative free (avoid computing Jacobian)
- global optimization (convergence in probability to global optimum)
- good convergence rate (provably asymptotic regret, $\mathcal{O}\left(n^{-\frac{1}{d}}\right)$)

Disadvantages:

- high-dimensionality
- scalability: computational bottleneck $\mathcal{O}(n^3)$ when $n \geq \mathcal{O}(10^3)$

Bayesian optimization features

Sandia National Laboratories

very versatile (open for methodological extensions)

- acquisition functions: PI, EI, UCB, Thompson sampling, entropy-based, KG, or combination among these
- constrained on objectives (known + unknown constraints) ✓
- multi-objective(Pareto frontier/optimal, domination) ✓
- multi-output ✓
- multi-fidelity ✓
- batch parallelization ✓ → asynchronous parallel ✓
- stochastic, heteroscedastic ✗
- time-series (forecasting, e.g. causal kernel) ✗
- mixed-integer (discrete/categorical + continuous) ✓
- scalable ✓
- latent variable model ✗
- gradient-enhanced ✓
- high-dimensional (with low effective dimensionality) ✓
- physics-constrained: monotonic, discontinuous, symmetric, bounded ✗
- outlier: student-t distribution ✗
- non-stationary kernels ✗

Classical GP: Fundamentals

Let $\mathcal{D}_n = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ denote the set of observations and $\mathbf{x}$ denote an arbitrary test points

$$\mu_n(\mathbf{x}) = \mu_0(\mathbf{x}) + \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}) \quad (1)$$

$$\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}) \quad (2)$$

where $\mathbf{k}(\mathbf{x})$ is a vector of covariance terms between $\mathbf{x}$ and $\mathbf{x}_{1:n}$.

Classical GP: Fundamentals

Formulation:

- assuming stationary $\rightarrow$ only depends on $r = ||\mathbf{x} - \mathbf{x}'||$
- the covariance matrix: symmetric positive-semidefinite matrix made up of pairwise inner products

$$\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i) = \mathbf{K}_{ji} \tag{3}$$

- kernel choice: assuming unknown function is smooth to some degree

Implementation:

- maximum log (marginal) likelihood estimation (MLE) to estimate the hyper-parameter $\theta \in \mathbb{R}^d$
- MLE involves $\mathbf{K}^{-1} \rightarrow \mathcal{O}(n^3)$
- size of $\mathbf{K} \in \mathbb{R}^{n \times n}$ increases as the optimization process advances

Ingredients: some data, GP kernel, acquisition function

Classical GP: Fundamentals

Matérn kernels:

$$\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) = \theta_0^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} r)^{\nu} K_{\nu}(\sqrt{2\nu} r), \tag{4}$$

K_ν is a modified Bessel fuction of the second kind and order ν.
Common kernels:

- $\nu = 1/2 : k_{\text{Matérn1}}(\mathbf{x}, \mathbf{x}') = \theta_0^2 \exp(-r)$ (also known as exponential kernel),
- $\nu = 3/2 : k_{\text{Matérn3}}(\mathbf{x}, \mathbf{x}') = \theta_0^2 \exp(-\sqrt{3}r)(1 + \sqrt{3}r)$,
- $\nu = 5/2 : k_{\text{Matérn5}}(\mathbf{x}, \mathbf{x}') = \theta_0^2 \exp(-\sqrt{5}r) \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right)$,
- $\nu \to \infty : k_{\text{sq-exp}}(\mathbf{x}, \mathbf{x}') = \theta_0^2 \exp\left(-\frac{r^2}{2}\right)$ (also known as square exponential or automatic relevance determination kernel)

Log-likelihood function:

$$\log p(\mathbf{y}|\mathbf{x}_{1:n}, \theta) = - \underbrace{\frac{n}{2}\log(2\pi)}_{\text{data likelihood } \downarrow \text{ as } n\uparrow} - \underbrace{\frac{1}{2}\log|\mathbf{K}^\theta + \sigma^2\mathbf{I}|}_{\substack{\text{“complexity” term} \\ \text{smoother covariance matrix}}} - \underbrace{\frac{1}{2}(\mathbf{y} - \mathbf{m}_\theta)^T(\mathbf{K}^\theta + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}_\theta)}_{\substack{\text{“data-fit” term} \\ \text{how well model fits data}}} \tag{5}$$

Classical GP: A Bayesian perspective

Mostly follow Quiñonero-Candela and Hansen 2004; Quiñonero-Candela and Rasmussen 2005.
Denote training $\mathbf{f}$, testing $\mathbf{f}_*$, the joint GP prior is

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{m} \\ \mathbf{m} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix} \right). \tag{6}$$

By Bayes' rule

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{y}) &= \int p(\mathbf{f}, \mathbf{f}_*|\mathbf{y}) d\mathbf{f} \\ &= \frac{1}{p(\mathbf{y})} \int p(\mathbf{y}|\mathbf{f})\ p(\mathbf{f}, \mathbf{f}_*) d\mathbf{f} \\ &= \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{f}}[\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}]^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}[\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}]^{-1}\mathbf{K}_{\mathbf{f},*}), \end{aligned} \tag{7}$$

Log of marginal likelihood function:

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}) &= \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{X}) d\mathbf{f} \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}| - \frac{1}{2} (\mathbf{y} - \mathbf{m})^\top (\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I})^{-1} (\mathbf{y} - \mathbf{m}). \end{aligned} \tag{8}$$

Classical GP: A Bayesian perspective

A conditional of a Gaussian is also Gaussian.

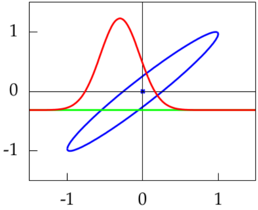


Figure 16: Photo courtesy of from Lawrence 2016.

If

$$P(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix}\mu_{\mathbf{x}} \\ \mu_{\mathbf{y}}\end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right) \tag{9}$$

then

$$P(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mu_{\mathbf{x}} + CB^{-1}(y - \mu_{\mathbf{y}}), A - CB^{-1}C^\top) \tag{10}$$

(cf. App. A, Quiñonero-Candela and Rasmussen 2005).

Acquisition function: How to pick the next point(s)

Sandia National Laboratories

- dictates how to pick the next point: exploitation (focus on the promising region) or exploration (focus on the uncertain/unknown region)
- different flavors:
  1. probability of improvement (PI) Mockus 1982

$$a_{\mathsf{PI}}(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta) = \Phi(\gamma(\mathbf{x})), \tag{11}$$

where

$$\gamma(\mathbf{x}) = \frac{\mu(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta) - f(\mathbf{x}_{\mathsf{best}})}{\sigma(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta)}, \tag{12}$$

  2. expected improvement (EI) scheme Mockus 1975; Huang et al. 2006

$$a_{\mathsf{EI}}(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta) = \sigma(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta) \cdot (\gamma(\mathbf{x})\Phi(\gamma(\mathbf{x})) + \phi(\gamma(\mathbf{x})) \tag{13}$$

Acquisition function: How to pick next point(s)

Sandia National Laboratories

- dictates how to pick the next point: exploitation (focus on the promising region) or exploration (focus on the uncertain/unknown region)
- different flavors:

3. upper confidence bound (UCB) schemeSrinivas et al. 2009; Srinivas et al. 2012

$$a_{\text{UCB}}(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta) = \mu(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta) + \kappa\sigma(\mathbf{x}; \{\mathbf{x}_i, y_i\}_{i=1}^n, \theta), \tag{14}$$

   where κ is a hyper-parameter describing the exploitation-exploration balance.

4. pure exploration*:
   - maximal MSE $\sigma^2(\mathbf{x}) \Leftrightarrow$ maximal entropy $\frac{1}{2}\log\left[2\pi\sigma^2(\mathbf{x})\right] + \frac{1}{2}$
   - maximal IMSE $\int_{\mathbf{x}\in\mathcal{X}} \sigma^2(\mathbf{x})$

QRAK taxonomy for constrained optimization problem

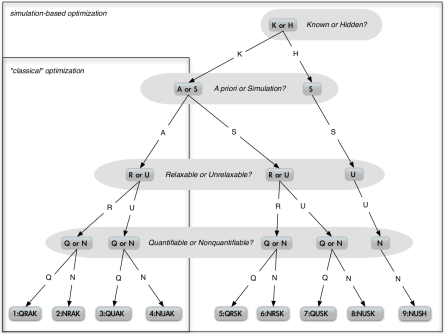


Figure 17: Photo courtesy of Digabel and Wild 2015. Tree-based view of the QRAK taxonomy of constraints. Constraints that are either not known beforehand or have to assessed through simulations are called unknown.

Constrained problems: known constraints

Problem statement

optimize $f(\mathbf{x})$ subject to $\lambda(\mathbf{x}) \leq \mathbf{c}$, $\lambda(\cdot)$ computationally cheap

known constraints:

- known before evaluation
- typically physics-based
- formulated as inequality constraints $\lambda(\mathbf{x}) \leq \mathbf{c}$, λ is computationally cheap
- directly penalize the acquisition function $a = 0$ when constraints are violated, i.e. $\lambda(\mathbf{x}) \nleq \mathbf{c}$

$$a_{\text{constrained}}^{\text{known}}(\mathbf{x}) = a(\mathbf{x}) I_{\text{known}}(\mathbf{x}) \tag{15}$$

where $I_{\text{known}}(\mathbf{x})$ is the indicator function

$$I_{\text{known}}(\mathbf{x}) = \begin{cases} 1, & \lambda(\mathbf{x}) \leq \mathbf{c} \\ 0, & \lambda(\mathbf{x}) \nleq \mathbf{c} \end{cases} \tag{16}$$

- can be conveniently ignored to become unknown constraints if the model is aware of the constraints violation, i.e. returns error

Constrained problems: unknown constraints

Problem statement

optimize $f(\mathbf{x})$ where $f(\mathbf{x})$ may or may not exist

unknown constraints:

- can convert known $\rightarrow$ unknown but not vice versa
- form a probabilistic binary classifier to predict the probability mass function of passing unknown constraint at $\mathbf{x}$, i.e. kNN, AdaBoost, RandomForest, GP, etc.
- penalize the acquisition function based on the predicted feasibility from GP classifier

$$a_{\text{constrained}}^{\text{unknown}}(\mathbf{x}) = \begin{cases} a(\mathbf{x}), & \text{with } \Pr(\text{clf}(\mathbf{x}) = 1) \\ 0, & \text{with } \Pr(\text{clf}(\mathbf{x}) = 0) \end{cases} \quad (17)$$

- our approach:
  - use another GP to learn when $f(\mathbf{x})$ does not exist
  - optimize the conditioned acquisition function $\mathbb{E}[a_{\text{constrained}}^{\text{unknown}}(\mathbf{x})] = a(\mathbf{x})\Pr_{\text{unknown}}(\text{clf}(\mathbf{x}) = 1)$

Batch parallel on HPC

Arguments:

- focus on multi-core HPC architecture and expensive, high-fidelity simulations
- Amdahl's law: diminishing returns, i.e. rewards for parallelizing solvers diminish as # of processors increase
- motivation: can we search for the optimal point in faster wall-clock time, assuming HPC power is sufficient and/or abundant?
- obviously beneficial when computing resource is sufficient

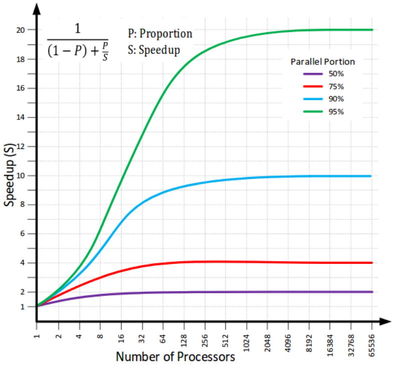


Figure 18: Amdahl's law for parallelization.

Batch parallel on HPC

Might as well be beneficial when computing resource is insufficient; examples:

- $P = 0.95 \rightarrow$ SpeedUp $\approx$ 20 times
- CFD simulation takes 3 hours to finish with 256 procs $\rightarrow$ 20 cases/60 hours
- or 60 hours (2.5 days) with 1 proc for 1 case $\rightarrow$ 256 cases/60 hours
- fixed computational budget: 256 $\times$60 CPU hours
- question: in the period of 2.5 days, are we better off with 20 sequential runs, or with 256 batch-parallel runs? what about 5 days (40 vs. 512)? 10 days (80 vs. 1024)? asymptotically?

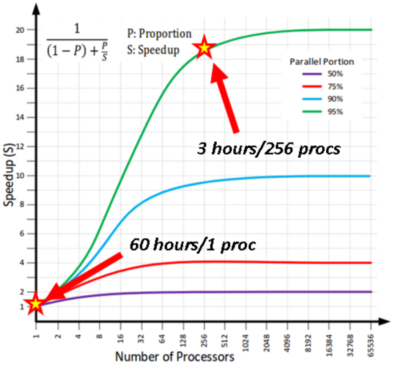


Figure 19: Amdahl's law for parallelization.

Batch parallel on HPC

Strategies to design batches:

- hallucination (GP-BUCB Desautels, Krause, and Burdick 2014): cast $\mu_{\text{predicted}} = \mu_{actual}$, $\sigma_{\text{predicted}} = 0$ during one iteration
  - 1 batches: all acquisition, zero exploration, UCB acquisition function
- pure exploration (GP-UCB-PE Contal et al. 2013)
  - 2 batches: 1 acquisition, the rest exploration, UCB acquisition function
- couple with unknown constraints: pBO-2GP-3B: 2 GPs, 3 batches: hallucination, exploration for GP_1 (main), exploration for GP_2 (classifier)
  - 3 batches: some acquisition, some exploration (GP_1), and some more exploration (GP_2);
  - all types of acquisition functions, dynamic batch settings are easily extended
  - order to construct the batch matters
- others: GP-KG Scott, Frazier, and Powell 2011; Wu and Frazier 2016, GP-SM Azimi, Fern, and Fern 2010, GP-DPP Kathuria, Deshpande, and Kohli 2016, GP-PPES Shah and Ghahramani 2015, GP-LP González et al. 2016, q-EI Marmin, Chevalier, and Ginsbourger 2016 Chevalier and Ginsbourger 2013, GP-Hedge Hoffman, Brochu, and Freitas 2011

Asynchronous parallel

A cartoonist's perspective

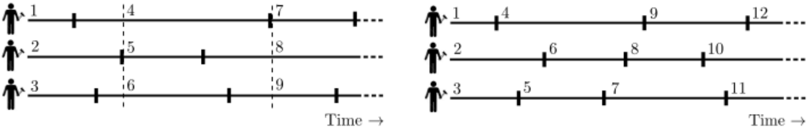


Figure 20: Synchronous (left) vs. asynchronous (right). Batch size = 3. Photo courtesy of Kandasamy et al Kandasamy et al. 2017.

Asynchronous parallel

... and a computer scientist's perspective

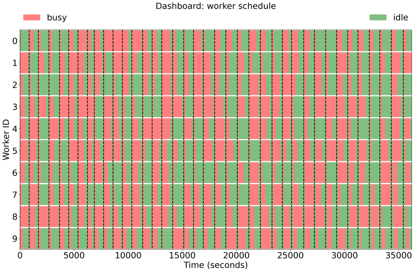


Figure 21: Batch-sequential parallel

Figure 22: Asynchronous parallel

Multi-fidelity

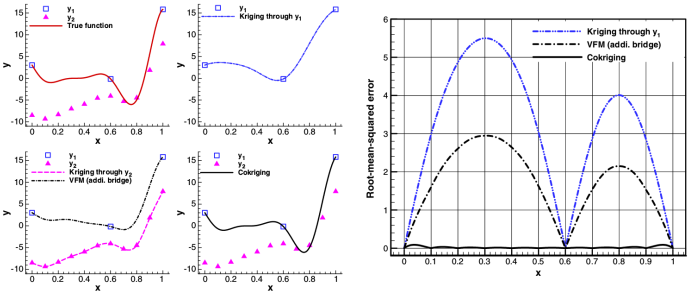


Figure 23: 1D approximation function from Forrester et al. Forrester, Sóbester, and Keane 2007, Han et al Solnik et al. 2017.

Multi-fidelity

Kennedy and O'Hagan Kennedy and O'Hagan 2000:
auto-regressive model based on a first-order auto-regressive relation between model output of different levels of fidelity.

- s-levels of variable-fidelity model $y_t(\mathbf{x})_{t=1}^{s}$
- $y_1(\mathbf{x})$: cheapest, $y_s(\mathbf{x})$: most expensive
- auto-regressive model:

$$y_t(\mathbf{x}) = \rho_{t-1} y_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}) \tag{18}$$

- Markov property: assuming that given $y_{t-1}(\mathbf{x})$, we can learn nothing about $y_t(\mathbf{x})$ from any other model output $y_{t-1}(\mathbf{x}')$, for $\mathbf{x} \neq \mathbf{x}'$

$$\mathrm{Cov}[y_t(\mathbf{x}), y_{t-1}(\mathbf{x}') | y_{t-1}(\mathbf{x})] = 0, \quad \forall \mathbf{x} \neq \mathbf{x}' \tag{19}$$

Multi-fidelity

- model the lowest fidelity y_1 as a classical GP
- model the discrepancies δ_t's as GPs
- for two levels of fidelity: $\blacksquare_c$ = cheap, $\blacksquare_e$ = expensive
- covariance vector and covariance matrix

$$k(\mathbf{x}) = (\rho\sigma_c^2 k_c(\mathbf{x}) \quad \rho\sigma^2 k_c(\mathbf{x}, \mathbf{X})), \tag{20}$$

$$\mathbf{K} = \begin{pmatrix} \sigma_c^2 \mathbf{K}_c & \rho\sigma_c^2 \mathbf{K}_c(\mathbf{X}_c, \mathbf{X}_e) \\ \rho\sigma_c^2 \mathbf{K}_c(\mathbf{X}_e, \mathbf{X}_c) & \rho^2\sigma_c^2 \mathbf{K}_c(\mathbf{X}_e, \mathbf{X}_e) + \sigma_d^2 \mathbf{K}_e(\mathbf{X}_e, \mathbf{X}_e) \end{pmatrix} \tag{21}$$

- low-fidelity MLE for θ_c; high-fidelity MLE for θ_e and ρ

$$\log p(\mathbf{y}_c|\mathbf{x}_{n_c}, \theta_c) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K_c}^{\theta_c} + \sigma_c^2\mathbf{I}| - \frac{1}{2}(\mathbf{y} - \mathbf{m}_{\theta_c})^T(\mathbf{K_c}^{\theta_c} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}_{\theta_c}) \tag{22}$$

$$\log p(\mathbf{y}_e|\mathbf{x}_{n_e}, \theta_e) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K_e}^{\theta_e} + \sigma_e^2\mathbf{I}| - \frac{1}{2}(\mathbf{y} - \mathbf{m}_{\theta_e})^T(\mathbf{K_e}^{\theta_e} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}_{\theta_e}) \tag{23}$$

Multi-fidelity

$y_s(\mathbf{x}) = \sum_{t=1}^{s-1} \rho_t y_t(\mathbf{x}) + \delta(\mathbf{x}) \Leftrightarrow \delta(\mathbf{x}) = y_s(\mathbf{x}) - \sum_{t=1}^{s-1} \rho_t y_t(\mathbf{x})$
Covariance matrix for s levels of fidelity Xiao et al. 2018

$$\mathbf{K} = \begin{pmatrix} \sigma_1^2 \mathbf{K}_1 & 0 & \cdots & \rho_1 \sigma_1^2 \mathbf{K}_1(\mathbf{X}_1, \mathbf{X}_e) \\ 0 & \sigma_2^2 \mathbf{K}_2 & \cdots & \rho_2 \sigma_2^2 \mathbf{K}_2(\mathbf{X}_2, \mathbf{X}_e) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_1 \sigma_1^2 \mathbf{K}_1(\mathbf{X}_e, \mathbf{X}_1) & \rho_2 \sigma_2^2 \mathbf{K}_2(\mathbf{X}_e, \mathbf{X}_2) & & \sum_{t=1}^{s} \rho_t^2 \sigma_t^2 \mathbf{K}_t(\mathbf{X}_e, \mathbf{X}_e) + \sigma_d^2 \mathbf{K}_e(\mathbf{X}_e, \mathbf{X}_e) \end{pmatrix} \tag{24}$$

MLE hyper-parameters optimization of ρ_t happens at the highest level of fidelity s, after all lower-fidelity hyper-parameters $\{\theta_t\}_{t=1}^{s-1}$ have been estimated.

Multi-fidelity

Variations in formulation:

- auto-regressive (Kennedy and O'Hagan Kennedy and O'Hagan 2000) vs. recursive with nested structure (Le Gratiet & Garnier Le Gratiet and Garnier 2014, Peridakis & Karniadakis Perdikaris et al. 2015; Perdikaris and Karniadakis 2016): $\mathcal{O}\left(\left(\sum n_t\right)^3\right)$ vs. $\sum \mathcal{O}\left((n_t)^3\right)$ by decomposing the covariance matrix
- nested (Le Gratiet & Garnier Le Gratiet and Garnier 2014, Peridakis & Karniadakis Perdikaris et al. 2015; Perdikaris and Karniadakis 2016) vs. non-nested (Couckuyt et al. Couckuyt et al. 2012; Couckuyt, Dhaene, and Demeester 2013; Couckuyt, Dhaene, and Demeester 2014, Xiao et al. Xiao et al. 2018)

Selection of level of fidelity

Question: Fix a sampling location $\mathbf{x}^*$, what level of fidelity should be selected to query?

Compare computational cost vs. benefit:

- $1 \leq t \leq s$: level of fidelity
- if $\mathbf{x}^*$ is queried, how much uncertainty is reduced?
- at what cost?
- balance computational cost vs. gain (reduction of uncertainty)

$$t^* = \underset{t}{\operatorname{argmin}} \left(C_t \int_{\mathcal{X}} \sigma^2(\mathbf{x}) d\mathbf{x} \right), \tag{25}$$

- promote high-fidelity if the cost is similar: If $C_{t^*} |\mathcal{D}^{(t^*)}| \geq C_s |\mathcal{D}^{(s)}|$ then choose s.

Multi-objective

Let:

- $\mathbf{x} = \{x_i\}_{i=1}^{d} \in \mathcal{X} \subseteq \mathbb{R}^d$ be input in d-dimensional space,
- $\mathbf{y} = \{y_j\}_{j=1}^{s}$ as s outputs.

$$\underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}(f_1(\mathbf{x}), \cdots, f_s(\mathbf{x})) \tag{26}$$

subjected to $\mathbf{c}(\mathbf{x}) \leq \mathbf{0}$.

Multi-objective

Pareto definition:

- $\mathbf{x}_1$ is said to dominate $\mathbf{x}_2$, denoted as $\mathbf{x}_1 \preceq \mathbf{x}_2$, if and only if $\forall 1 \leq j \leq s$, such that $y_j(\mathbf{x}_1) \leq y_j(\mathbf{x}_2)$, and $\exists 1 \leq j \leq s$, such that $y_j(\mathbf{x}_1) < y_j(\mathbf{x}_2)$.
- $\mathbf{x}_1$ is said to strictly dominate $\mathbf{x}_2$, denoted as $\mathbf{x}_1 \prec \mathbf{x}_2$, if and only if $\forall 1 \leq j \leq s$, such that $y_j(\mathbf{x}_1) < y_j(\mathbf{x}_2)$.

Scalarization: multi-objective $\rightarrow$ single-objective

1. weighted Tchebycheff $y = \max_{1 \leq i \leq s} w_i(y_i(\mathbf{x}) - z_i^*)$,
2. weighted sum $y = \sum_{i=1}^{s} w_i y_i(\mathbf{x})$,
3. augmented Tchebycheff
   $y = \max_{1 \leq i \leq s} w_i(y_i(\mathbf{x}) - z_i^*) + \rho \sum_{i=1}^{s} w_i y_i(\mathbf{x})$,

where z_i^* denotes the ideal value for the i-th objective, the weights $0 \leq w_i \leq 1$, $\sum_{i=1}^{m} w_i = 1$, ρ is a positive constant ($\rho = 0.05$).

Multi-objective

Acquisition function:

$$a(\mathbf{x}) = \underbrace{a_{\text{obj}}(\mathbf{x})}_{\text{objective GP}} \cdot \underbrace{a_{\text{Pareto}}(\mathbf{x})}_{\text{uncertain Pareto}} \cdot \underbrace{\Pr(\mathbf{x}|c(\mathbf{x}) = 1)}_{\text{unknown constraints}} \cdot \underbrace{\mathcal{I}(\mathbf{x})}_{\text{known constraints}} \quad (27)$$

- $a_{\text{obj}}(\mathbf{x})$: objective GP fitted through augmented Tchebycheff with random weights
- $a_{\text{Pareto}}(\mathbf{x})$: Pareto GP classifier (Pareto/non-Pareto)
- $\Pr(\mathbf{x}|c(\mathbf{x}) = 1)$: constrained classifier (feasible/infeasible)
- $\mathcal{I}(\mathbf{x})$: indicator function if $\mathbf{c}(\mathbf{x}) \leq \mathbf{0}$

Multi-objective

Hypervolume approach:

- hypervolume indicator, aka $\mathcal{S}$-metric
- strictly monotonic
- complexity $\mathcal{O}(n \log n + n^{d/2} \log n)$
- $d = 3$: lower and upper bounds $\mathcal{O}(n \log n)$ Beume et al. 2009
- and any other sorts of approximation ...
- arguably more sample-efficient compared to Tchebycheff decomposition

Sparse GP

Low-rank approximation[1]for $\mathbf{K}_{\mathbf{f},\mathbf{f}}$

Low-rank approximation $\mathbf{K} \approx \widetilde{\mathbf{K}} = \mathbf{K}_{n\times m}\mathbf{K}_{m\times m}^{-1}\mathbf{K}_{m\times n}$ (cf. Section 8.1 Rasmussen 2006) and scales as $\mathcal{O}(nm^2 + m^3)$ instead of $\mathcal{O}(n^3)$.
For $n \gg m$, this method scales as $\mathcal{O}(nm^2)$.

Following Quiñonero-Candela and Rasmussen 2005; Quiñonero-Candela, Rasmussen, and Williams 2007, Chalupka, Williams, and Murray 2013, Vanhatalo et al. 2012; Vanhatalo et al. 2013.
Cost complexity:

- local GP: $\mathcal{O}(m^3)$
- sparse GP: $\mathcal{O}(nm^2)$
- classical GP (Cholesky decomposition): $\mathcal{O}\left(\frac{1}{3}n^3\right)$
- classical GP (LU decomposition): $\mathcal{O}\left(\frac{2}{3}n^3\right)$
- classical GP (QR decomposition): $\mathcal{O}\left(\frac{4}{3}n^3\right)$

[2]https://en.wikipedia.org/wiki/Low-rank_matrix_approximations

Sparse GP

Sandia National Laboratories

- $p(\cdot)$: true pdf
- $q(\cdot)$: approximate pdf

Assume the fully independent training conditional (FITC) Quiñonero-Candela and Rasmussen 2005; Quiñonero-Candela, Rasmussen, and Williams 2007, augment the joint model $p(\mathbf{f}_*, \mathbf{f})$ as

$$p(\mathbf{f}_*, \mathbf{f}) = \int p(\mathbf{f}_*, \mathbf{f}, \mathbf{u}) d\mathbf{u} = \int p(\mathbf{f}_*, \mathbf{f}|\mathbf{u}) p(\mathbf{u}) d\mathbf{u}, \tag{28}$$

$\mathbf{u}$: inducing variables at m locations $\mathbf{X}_\mathbf{u}$. The training and testing conditionals are

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{\mathbf{f},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} (\mathbf{u} - \mathbf{m}),\ \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}}), \tag{29}$$

and

$$p(\mathbf{f}_*|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} (\mathbf{u} - \mathbf{m}),\ \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}), \tag{30}$$

where

$$\mathbf{Q}_{\mathbf{a},\mathbf{b}} := \mathbf{K}_{\mathbf{a},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{b}}. \tag{31}$$

The likelihood and inducing priors remain the same, i.e. $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I})$, and $p(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{K}_{\mathbf{u},\mathbf{u}})$.

Sparse GP

FITC training prior based on the inducing priors is modified as

$$q(\mathbf{f}|\mathbf{u}) = \prod_{i=1}^{n} p(\mathbf{f}_i|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}(\mathbf{u}-\mathbf{m}), \text{Diag}[\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}}]) \tag{32}$$

and keeping the testing prior the same

$$q(\mathbf{f}_*|\mathbf{u}) = p(\mathbf{f}_*|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}(\mathbf{u}-\mathbf{m}),\ \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}), \tag{33}$$

the effective prior under the FITC assumption is

$$q(\mathbf{f},\mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix}\mathbf{m}\\ \mathbf{m}\end{bmatrix}, \begin{bmatrix}\mathbf{Q}_{\mathbf{f},\mathbf{f}} - \text{Diag}[\mathbf{Q}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{f}}] & \mathbf{Q}_{\mathbf{f},*}\\ \mathbf{Q}_{*,\mathbf{f}} & \mathbf{K}_{*,*}\end{bmatrix}\right), \tag{34}$$

which implies the testing distribution as

$$\begin{aligned} q(\mathbf{f}_*|\mathbf{y}) &= \mathcal{N}(\mathbf{m} + \mathbf{Q}_{*,\mathbf{f}}(\mathbf{Q}_{\mathbf{f},\mathbf{f}} + \Lambda)^{-1}(\mathbf{y}-\mathbf{m}), \mathbf{K}_{*,*} - \mathbf{Q}_{*,\mathbf{f}}(\mathbf{Q}_{\mathbf{f},\mathbf{f}} + \Lambda)^{-1}\mathbf{Q}_{\mathbf{f},*}) \\ &= \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{u}}\Sigma\mathbf{K}_{\mathbf{u},\mathbf{f}}\Lambda^{-1}(\mathbf{y}-\mathbf{m}), \mathbf{K}_{*,*} - \mathbf{Q}_{*,*} + \mathbf{K}_{*,\mathbf{u}}\Sigma\mathbf{K}_{\mathbf{u},*}) \end{aligned}, \tag{35}$$

where $\Sigma = [\mathbf{K}_{\mathbf{u},\mathbf{u}} + \mathbf{K}_{\mathbf{u},\mathbf{f}}\Lambda^{-1}\mathbf{K}_{\mathbf{f},\mathbf{u}}]^{-1}$ and $\Lambda = \text{Diag}[\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}]$.

Sparse GP

The marginal likelihood conditioned on the inducing inputs is therefore

$$q(\mathbf{y}|\mathbf{X_u}) = \int\int p(\mathbf{y}|\mathbf{f})q(\mathbf{f}|\mathbf{u})p(\mathbf{u}|\mathbf{X_u})d\mathbf{u}d\mathbf{f} = \int p(\mathbf{y}|\mathbf{f})q(\mathbf{f}|\mathbf{X_u})d\mathbf{f}, \quad (36)$$

which implies the log marginal likelihood as

$$\log q(\mathbf{y}|\mathbf{X_u}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{Q_{f,f}} + \Lambda| - \frac{1}{2}(\mathbf{y}-\mathbf{m})^\top[\mathbf{Q_{f,f}} + \Lambda]^{-1}(\mathbf{y}-\mathbf{m}), \quad (37)$$

where $\Lambda = \text{Diag}[\mathbf{K_{f,f}} - \mathbf{Q_{f,f}}] + \sigma^2\mathbf{I}$.
Cost complexity: $\mathcal{O}(nm^2)$ Williams and Seeger 2001; Li, Kwok, and Lü 2010. (Note: do not multiply matrices directly – cf. Section 14.3 Martinsson and Tropp 2020).

Variational inference: a hand-waving argument

Sandia National Laboratories

Follows Frigola, Chen, and Rasmussen 2014 and Rasmussen's corresponding slides. By Bayes' rule,

$$p(\mathbf{f}|\mathbf{y},\theta) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{p(\mathbf{y}|\theta)} \Leftrightarrow p(\mathbf{y}|\theta) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{p(\mathbf{f}|\mathbf{y},\theta)}. \quad (38)$$

The idea: approximate the (computationally intractable) $p(\mathbf{f}|\mathbf{y},\theta)$ by a (computationally tractable) parameterized variational $q(\mathbf{f})$. For any $q(\mathbf{f})$,

$$p(\mathbf{y}|\theta) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{p(\mathbf{f}|\mathbf{y},\theta)}\frac{q(\mathbf{f})}{q(\mathbf{f})} \Leftrightarrow \log p(\mathbf{y}|\theta) = \log\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{q(\mathbf{f})} + \log\frac{q(\mathbf{f})}{p(\mathbf{f}|\mathbf{y},\theta)}. \quad (39)$$

Apply $\int q(\mathbf{f})d\mathbf{f}$ to both sides

$$\underbrace{\log p(\mathbf{y}|\theta)}_{\text{marginal likelihood}} = \underbrace{\int q(\mathbf{f})\log\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{q(\mathbf{f})}d\mathbf{f}}_{\text{Evidence Lower BOund}} + \underbrace{\int q(\mathbf{f})\log\frac{q(\mathbf{f})}{p(\mathbf{f}|\mathbf{y},\theta)}d\mathbf{f}}_{KL(q(\mathbf{f})||p(\mathbf{f}|\mathbf{y},\theta))} \quad (40)$$

Turn our attention to maximizing the variational ELBO (or equivalently minimizing the KL divergence) instead of maximizing the log marginal likelihood.

High-dimensional: Gaussian random projection

Mostly follow Wang et al. 2013; Wang et al. 2016. Main idea:

- choose (wisely) and optimize over $\mathcal{Z} \subset \mathbb{R}^d$
- embed and project onto high-dimensional space as $\mathbf{x} \leftarrow p_{\mathcal{X}}(\mathbf{A}z)$
- $\mathbf{A} \in \mathbb{R}^{D \times d}$: tall-and-skinny random matrix with standard normal element

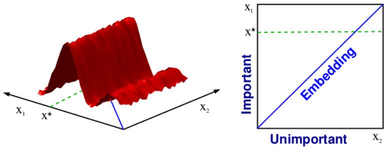


Figure 24: Photo courtesy of Wang et al Wang et al. 2016. Optimizing a 2d function (with 1d active subspace) via random embedding.

REMBO algorithm Wang et al. 2016 with deviation from BO highlighted.

generate a random matrix $\mathbf{A} \in \mathbb{R}^{D \times d} : a_{ij} \sim \mathcal{N}(0, 1)$
choose the bounded region set $\mathcal{Z} \subset \mathbb{R}^d$
$\mathcal{D}_0 \leftarrow \varnothing$
for $i = 1, 2, \cdots$ **do**
    locate next sampling point $\mathbf{z}_{i+1} \leftarrow \mathrm{argmax}_{\mathbf{z} \in \mathcal{Z}}\, a(\mathbf{z}) \in \mathbb{R}^d$
    query $\mathcal{D}_{i+1} \leftarrow \mathcal{D}_i \cup \{z_{i+1}, f(p_{\mathcal{X}}(\mathbf{A}z_{i+1}))\}$
    update GP
end for

High-dimensional: Gaussian random projection[2]

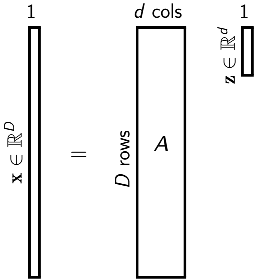


Figure 25: A random embedding or a random projection $\mathbf{x} = \mathbf{A}\mathbf{z}$ is built as a corollary from the Johnson-Lindenstrauss lemma, where $\mathbf{A}$ is a random normal matrix.

Theorem (Johnson-Lindenstrauss lemma (cf. Lemma 15 Mahoney 2016))

Given n points $\{\mathbf{x}_i\}_{i=1}^n$, each of which is in $\mathbb{R}^D$, $\mathbf{A} \sim \mathcal{MN}_{D\times d}(0, \mathbf{I}, \mathbf{I})$, and let $\mathbf{z} \in \mathbb{R}^d$ defined as $\mathbf{z} = \mathbf{A}^\top \mathbf{x}$. Then, if $d \geq \frac{9 \log n}{\varepsilon^2 - \varepsilon^3}$, for some $\varepsilon \in \left(0, \frac{1}{2}\right)$, then with probability at least $\frac{1}{2}$, all pairwise distances are preserved, i.e. for all i, j, we have

$$(1-\varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \leq \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \leq (1+\varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \quad (41)$$

Compared to active subspace method: also linear and does not require gradient and the rotation matrix $\mathbf{W}^\top$.
There are alternative approaches, e.g. additive GP.

[2] https://en.wikipedia.org/wiki/Random_projection

Mixed-integer

Main idea: (1) decompose to a set of continuous and discrete variables $\mathbf{x} = (\mathbf{x}_d, \mathbf{x}_c)$, (2) enumerate clusters, then (3) combine using linear weighted average

- decompose a large dataset into smaller clusters
- build a GP for each cluster
- each cluster corresponds to a unique tuple of discrete variables
- formulate a Gaussian mixture prediction: weighted average (weight tuned adaptively by statistical metrics, e.g. Wasserstein distance, Manhattan distance)
- applicable when $|\mathbf{x}_c| \gg |\mathbf{x}_d|$, i.e. not combinatorial optimization problems

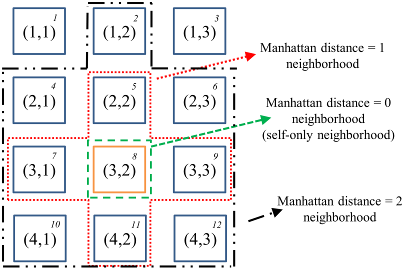


Figure 26: Neighborhood $\mathcal{B}(\ell)$ of a local GP ℓ with $\mathbf{x}_d = (3, 2)$.

Mixed-integer

Gaussian mixture model predictions for posterior mean and variance:

$$\hat{\mu} = \sum_{\ell^* \in \mathcal{B}(\ell)} w_{\ell^*} \Bigg(\hat{\mu}^{(\ell^*)} + \underbrace{\bar{\mu}^{(\ell)} - \bar{\mu}^{(\ell^*)}}_{\substack{\text{bias correction term} \\ \mathbb{E}[\hat{\mu}]=\bar{\mu}^{(\ell)}}} \Bigg) \quad (42)$$

$$\hat{\sigma}^2 = \sum_{\ell^* \in \mathcal{B}(\ell)} w_{\ell^*}^2 \sigma_{(\ell^*)}^2 \quad (43)$$

- $\mathcal{B}(\ell)$ is the neighborhood, defined by thresholding a similarity measure of discrete tuples
- weighted average estimation, weights depends on (1) cluster distances, (2) original cluster predictions
- theoretical bounds for weighted average prediction
- asymptotic behavior when $n \to \infty$

Mixed-integer

A special case: Wasserstein distance (Earth mover's distance). Assume the query point $\mathbf{x} = (\mathbf{x}_d, \mathbf{x}_c)$, where $\mathbf{x}_d$ corresponds to ℓ-th cluster.

$$w_{\ell*} \propto \left[\sigma_l^2 + W_2\left(\mathcal{N}(y^{(\ell^*)}, \sigma^2_{(\ell*)}), \mathcal{N}(y^{(\ell)}, \sigma^2_{(\ell)})\right)\right]^{-1}. \quad (44)$$

$$W_2\left(\mathcal{N}(y^{(\ell^*)}, \sigma^2_{(\ell*)}), \mathcal{N}(y^{(\ell)}, \sigma^2_{(\ell)})\right) = \left\|y^{(\ell)} - y^{(\ell^*)}\right\|^2 + \left\|\sqrt{\sigma^2_{(\ell)}} - \sqrt{\sigma^2_{(\ell*)}}\right\|^2 \quad (45)$$

Weighted prediction

The largest weight is associated with the ℓ-th cluster.

Asymptotic analysis $n \to \infty$

$\lim_{n\to\infty} w_l \to \infty$, as $\sigma_l \to 0$ and $W_2(\cdot_l, \cdot_l) = 0$.

Interpretation: If data is abundant, then the proposed approach converge asymptotically to a single local GP prediction.

Mixed-integer

Sandia National Laboratories

Simple bounds

The predicted mean $\hat{\mu} = \sum_{\ell^* \in \mathcal{B}(\ell)} w_{\ell^*} \left(\hat{\mu}^{(\ell^*)} + \bar{\mu}^{(\ell)} - \bar{\mu}^{(\ell^*)}\right)$ is bounded by

$$\min_{\ell^*} \left(\hat{\mu}^{(\ell^*)} + \bar{\mu}^{(\ell)} - \bar{\mu}^{(\ell^*)}\right) \leq \hat{\mu} \leq \max_{\ell^*} \left(\hat{\mu}^{(\ell^*)} + \bar{\mu}^{(\ell)} - \bar{\mu}^{(\ell^*)}\right) \tag{46}$$

The predicted variance $\hat{\sigma}^2 = \sum_{\ell^* \in \mathcal{B}(\ell)} w_{\ell^*}^2 \sigma_{(\ell^*)}^2$ is bounded by

$$\left(\sum_{\ell^*} w_{\ell^*}^2 \sigma_{(\ell^*)}\right)^2 \leq \hat{\sigma}^2 \leq \max_{\ell^*} \sigma_{(\ell^*)}^2 \tag{47}$$

Gaussian process / Bayesian optimization

Multi-scale engineering applications

- Numerical functions (analytical)
- Flip-chip BGA package design (FEM)
- Heart valve optimization (FEM)
- Metamaterials (FEM)
- Pump design optimization (CFD)
- AM process-structure modeling (kinetic Monte Carlo)
- Random ternary alloy composition design (DFT + MD/ML-IAP)
- Infer microstructure distribution (CPFEM)

Conclusion

References

2d three-hump camel

(parallel blind constraints)

(joint work w/ Yan Wang)

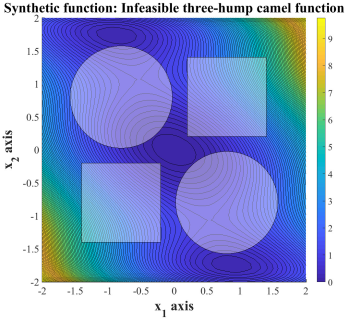


Figure 27: 2d three-hump camel.

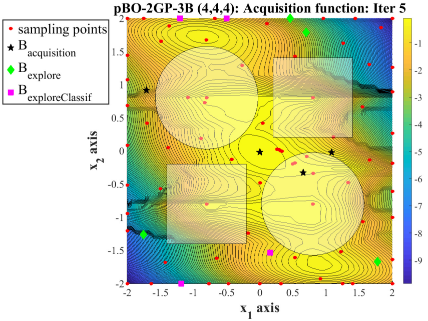


Figure 28: Batch sampling location at iteration 5.

2d three-hump camel

(parallel blind constraints)

(joint work w/ Yan Wang)

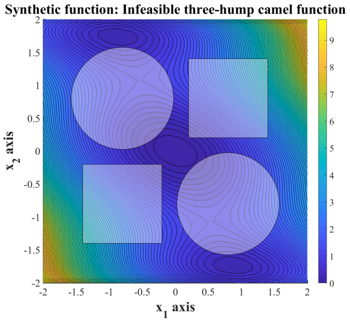


Figure 29: 2d three-hump camel.

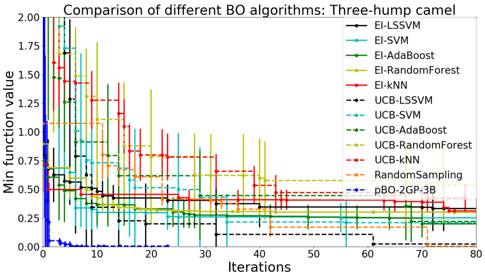


Figure 30: Convergence comparison with different classifiers.

2d Rastrigin

(parallel blind constraints)

(joint work w/ Yan Wang)

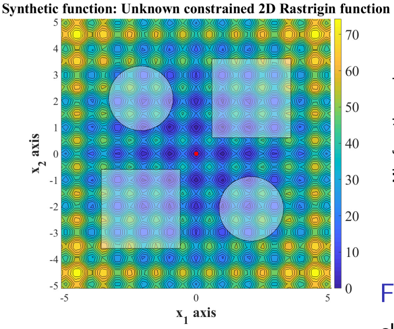


Figure 31: 2d three-hump camel.

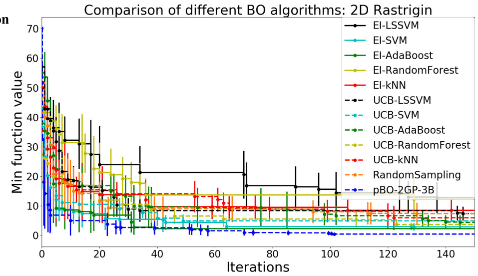


Figure 32: Convergence comparison with different classifiers.

6d Rastrigin

(parallel blind constraints)

(joint work w/ Yan Wang)

$g_i(\mathbf{x}) = \|\mathbf{x} - 2.56\mathbf{v}_i\|_2 \geq 5$, for $i = 1, \cdots, 6, \mathbf{v}_i = [-1, \cdots, 1 \cdots, -1]$ is a vector where i-index element is 1, and other elements are -1.

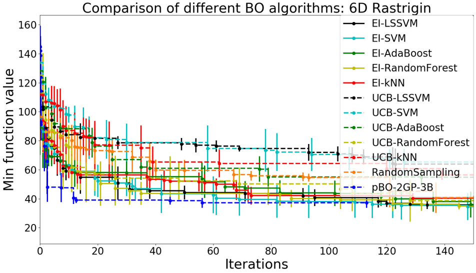


Figure 33: Convergence comparison with different classifiers.

Welded-beam design optimization

(2d+4d) (mixed-integer)

(joint work w/ Yan Wang)

Sandia National Laboratories

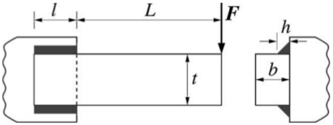


Figure 34: Welded-beam design

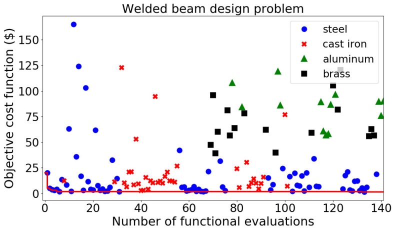


Figure 35: Convergence plot of welded beam design.

Speed reducer design optimization

(1d+6d) (mixed-integer)

(joint work w/ Yan Wang)

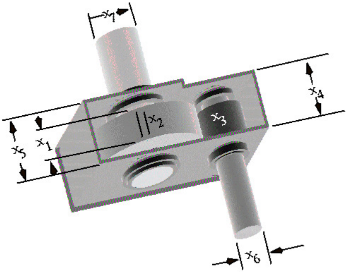


Figure 36: Speed reducer design

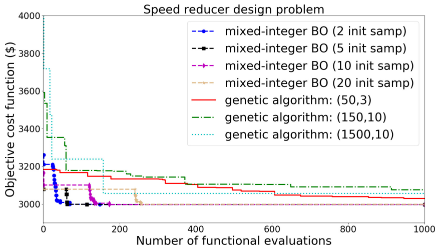


Figure 37: Comparison against GA.

High-dimensional discrete sphere function

(5d+50d) (mixed-integer)

(joint work w/ Yan Wang)

$f(\mathbf{x}^{(d)}, \mathbf{x}^{(c)}) = f(x_1, \cdots, x_n, x_{n+1}, \cdots, x_m) = \prod_{i=1}^{n} |x_i| \left(\sum_{j=n+1}^{m} x_j^2 \right)$ where $1 \leq x_i \leq 2 (1 \leq i \leq n)$ are n integer variables and $-5.12 \leq x_j \leq 5.12 (n+1 \leq j \leq m)$ are $m - n$ continuous variables.

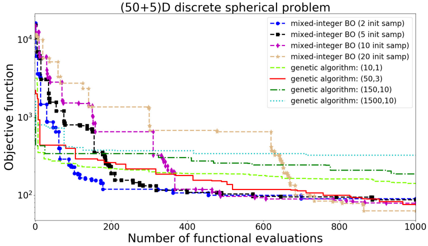


Figure 38: Comparison against GA.

High-dimensional discrete sphere function

(5d+100d) (mixed-integer)

(joint work w/ Yan Wang)

$f(\mathbf{x}^{(d)}, \mathbf{x}^{(c)}) =$
$f(x_1, \cdots, x_n, x_{n+1}, \cdots, x_m) =$
$\prod_{i=1}^{n} |x_i| \left(\sum_{j=n+1}^{m} x_j^2 \right)$
where
$1 \leq x_i \leq 2 (1 \leq i \leq n)$ are
n integer variables and
$-5.12 \leq x_j \leq$
$5.12 (n+1 \leq j \leq m)$ are
$m - n$ continuous
variables.

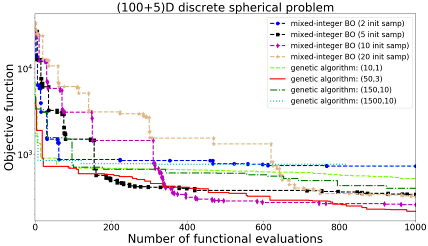


Figure 39: Comparison against GA.

Multi-objective: 2 objectives

(joint work w/ Mike Eldred)

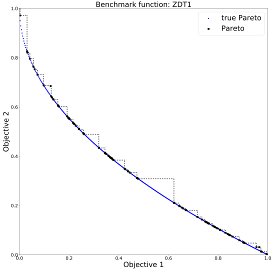


Figure 40: ZDT1.

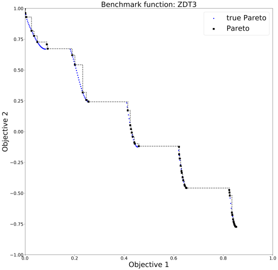


Figure 41: ZDT3.

Multi-objective: 3 objectives

(joint work w/ Mike Eldred)

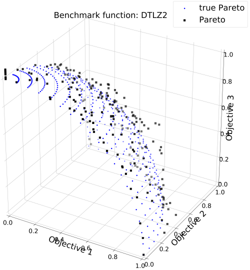


Figure 42: DTLZ2.

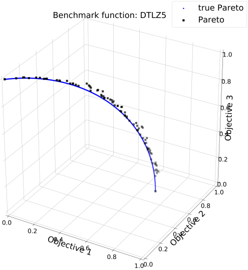


Figure 43: DTLZ5.

Multi-fidelity: borehole8d

(joint work w/ Scott McCann, Tim Wildey)

$$f_H(\boldsymbol{x}) = \frac{2\pi x_3(x_4 - x_6)}{\log(x_2/x_1)\left(1 + \frac{2x_7x_3}{\log(x_2/x_1)x_1^2x_8} + \frac{x_3}{x_5}\right)}, \quad (48)$$

$$f_L(\boldsymbol{x}) = \frac{5x_3(x_4 - x_6)}{\log(x_2/x_1)\left(1.5 + \frac{2x_7x_3}{\log(x_2/x_1)x_1^2x_8} + \frac{x_3}{x_5}\right)}. \quad (49)$$

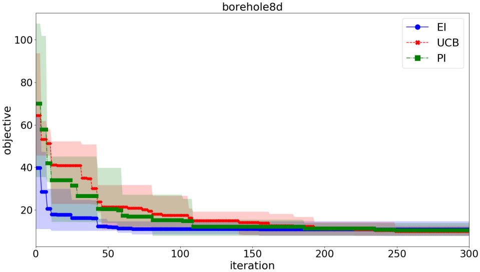


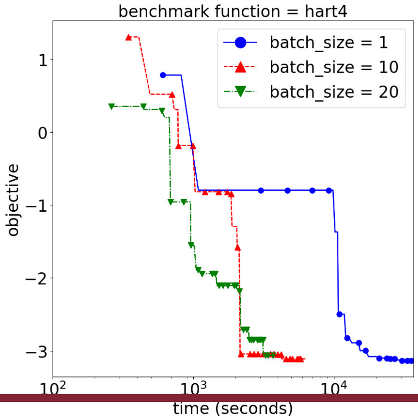
Figure 44: Borehole function (8d) - 2 levels of fidelity.

Asynchronous parallel

(joint work w/ Mike Eldred)

Hart4 function, $t \sim \mathcal{U}[30, 900]$s

$$f(\mathbf{x}) = \frac{1}{0.839}\left[1.1 - \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=4}^{3} A_{ij}(x_j - P_{ij})^2\right)\right], \tag{50}$$



Sparse GP for Big Data

(joint work w/ Bart G van Bloemen Waanders)

- Intel Xeon Platinum 8160 CPU @ 2.10GHz
- 24 cores, 48 threads
- RHEL 7.1 (Maipo)
- 180 GB of memory
- sphere function $y = \left(\sum_{i=1}^{3} x_i\right)^2$, $\mathcal{X} = [-1, 1]^3$
- training data points: $n \in \{10^1, 10^2, \ldots, 10^6\}$
- number of inducing points: $m \in \{10, 50, 100, \ldots, 300\}$
- GPstuff with SuitSparse toolbox on MATLAB
- $m = 300$, $n = 10^6$ takes ∼48 minutes

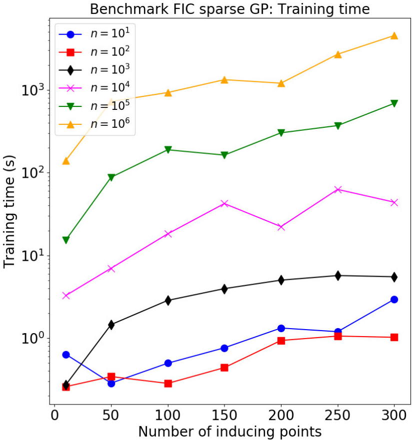


Figure 45: Benchmark of training time.

Sparse GP for Big Data

(joint work w/ Bart G van Bloemen Waanders)

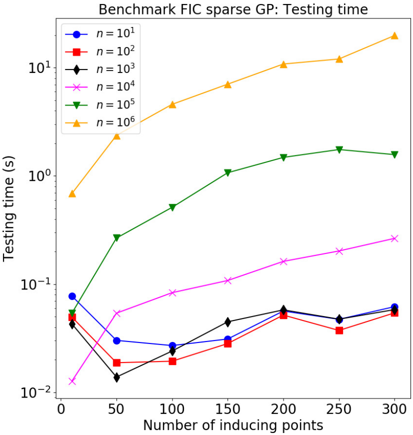


Figure 46: Benchmark of testing time.

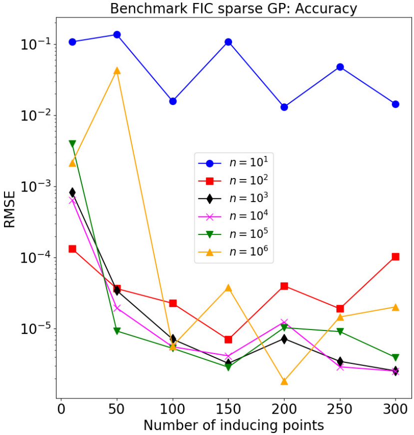


Figure 47: Benchmark of accuracy.

High-dimensional (with low effective dimensionality): Gaussian random projection

(joint work w/ Bart G van Bloemen Waanders)

The modified ZDT1 function, which is defined on $[-1, 1]^D$, is

$$f_2(\mathbf{x}) = g\left(1 - \sqrt{\frac{x_1^2}{g}}\right), \qquad (51)$$

where $g = 1 + 9\left(\sum_{i=2}^{D} \frac{x_i}{D-1}\right)^2$.

- (non-unique) global minimizer $\mathbf{x}^* = [1, 0, \ldots, 0]$
- $f_2(\mathbf{x}^*) = 0$
- $D = 10^4$
- $d = 10$
- $d_e = 2$

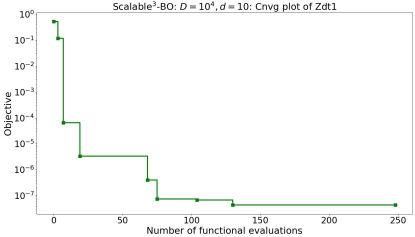


Figure 48: Convergence plot with $D = 10,000$, $d = 10$.

High-dimensional (with low effective dimensionality): Gaussian random projection

(joint work w/ Bart G van Bloemen Waanders)

The modified ZDT2 function, which is defined on $[-1, 1]^D$, is

$$f_2(\mathbf{x}) = g \left[1 - \left(\frac{x_1}{g} \right)^2 \right], \qquad (52)$$

where $g = 1 + \left(9 \sum_{i=2}^{D} x_i \right)^2$.

- (non-unique) global minimizer $\mathbf{x}^* = [1, 0, \ldots, 0]$
- $f_2(\mathbf{x}^*) = 0$
- $D = 10^4$
- $d = 3$
- $d_e = 2$

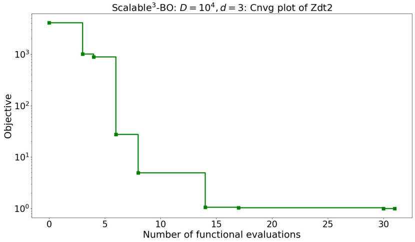
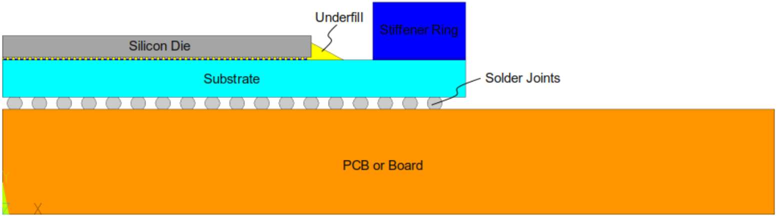


Figure 49: Convergence plot with $D = 10,000$, $d = 3$.

Flip-chip BGA package design (multi-fidelity BO + FEM)

(joint work w/ Scott McCann (Xilinx))

Figure 50: FE model geometry



- 2.5D FE on (ANSYS) APDL: half symmetry to reduce comp. time
- evaluate component warpage at 20°C and 200°C, and the strain energy density to predict the fatigue life of the solder joints during thermal cycling
- two levels of fidelity: varies mesh density parameter
- average comp. time: 0.4 CPU hr for low-fidelity, $\sim$ 1 CPU hr for high-fidelity

Flip-chip BGA package design (multi-fidelity BO + FEM)

(joint work w/ Scott McCann (Xilinx))

Table 1: Design variables for the FCBGA design optimization.

Variable	**Design part**	**Lower bound**	**Upper bound**	**Optimal value**
x_1	die	20000	30000	20702
x_2	die	300	750	320
x_3	substrate	30000	40000	35539
x_4	substrate	100	1800	1614
x_5	substrate	$10 \cdot 10^{-6}$	$17 \cdot 10^{-6}$	$17 \cdot 10^{-6}$
x_6	stiffener ring	2000	6000	4126
x_7	stiffener ring	100	2500	1646
x_8	stiffener ring	$8 \cdot 10^{-6}$	$25 \cdot 10^{-6}$	$8.94 \cdot 10^{-6}$
x_9	underfill	1.0	3.0	1.52
x_{10}	underfill	0.5	1.0	0.804
x_{11}	PCB board	$12.0 \cdot 10^{-6}$	$16.7 \cdot 10^{-6}$	$16.7 \cdot 10^{-6}$

Flip-chip BGA package design (multi-fidelity BO + FEM)

(joint work w/ Scott McCann (Xilinx))

Figure 51: Warpage at -40°C

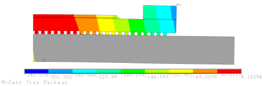


Figure 52: Warpage at 20°C

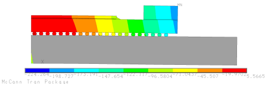
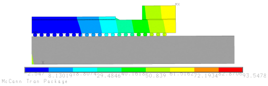


Figure 53: Warpage at 200°C



Flip-chip BGA package design (multi-fidelity BO + FEM)

(joint work w/ Scott McCann (Xilinx))

Figure 54: FE model

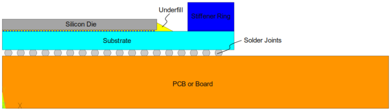
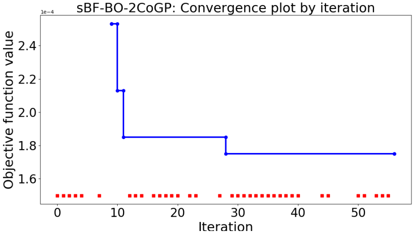


Figure 55: Conv. plot at high-fidelity



Heart valve optimization (FEM)

(joint work w/ Yan Wang, Wei Sun)

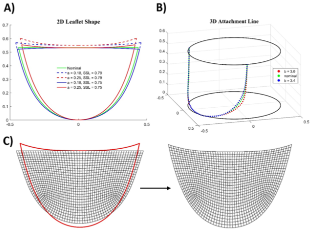


Figure 56: (A) Parameterization of 2D leaflet geometry; (B) 3D attachment edge shape; (C) Template leaflet mesh and nodes transformation.

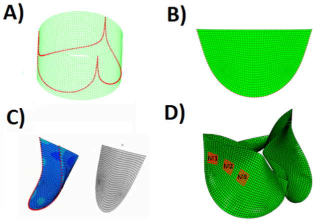


Figure 57: (A) 3D suturing line; (B) 2D attachment edge; (C) 2D-to-3D transformation; (D) Node and element mid-leaflet sets.

Mechanical metamaterials/AM

(joint work w/ Yan Wang)

Figure 58: Hierarchical multiscale structure of octahedral (second-order). Printed in Georgia Tech Invention Studio.

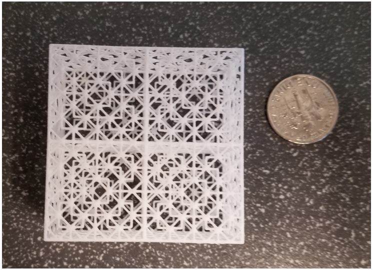


Figure 59: Design optimization of fractal cube. Printed in Georgia Tech Invention Studio.

Mechanical metamaterials/AM

(joint work w/ Yan Wang)

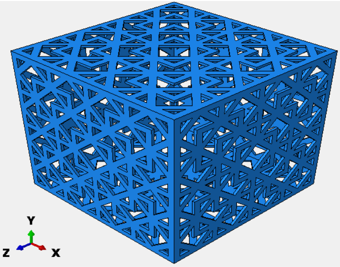


Figure 60: Parametric design.

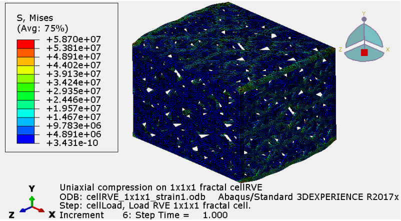


Figure 61: ABAQUS FEM.

(Fractal) auxetic metamaterials

(joint work w/ Yan Wang)

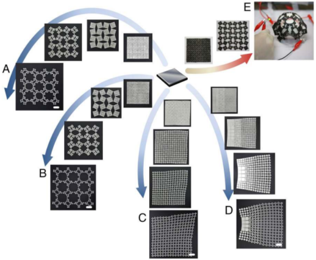


Figure 62: Stretchable electrode. Photo courtesy of Cho et al Cho et al. 2014.

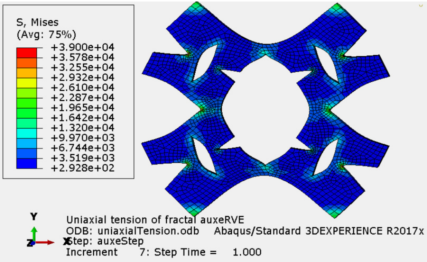


Figure 63: Application on auxetic (negative Poisson ratio) metamaterials.

Impeller design optimization using CFD

(joint work w/ GIW Industries)

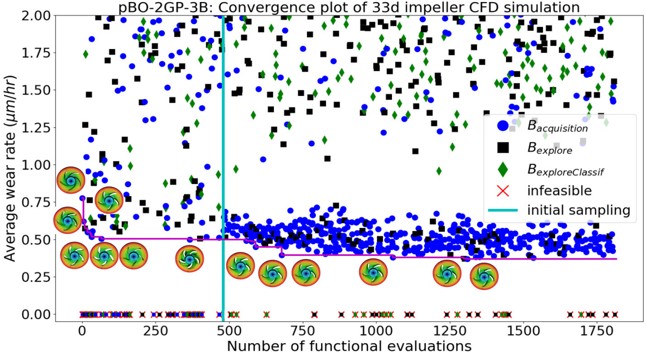


Figure 64: Multiphase CFD simulation for design optimization of 33d slurry pump impeller: Convergence plot.

Materials Design

process-structure-property linkage in materials

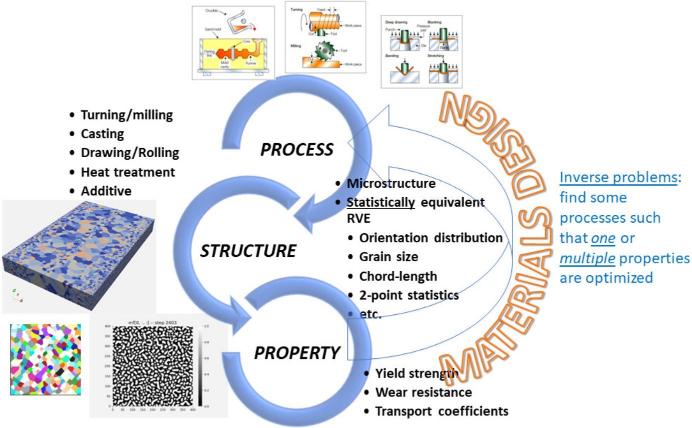


Figure 65: Process-Structure-Property linkage.

AM with kinetic Monte Carlo (AM/kMC)

(joint work w/ Laura Swiler, John Mitchell, Tim Wildey, Theron Rodgers)

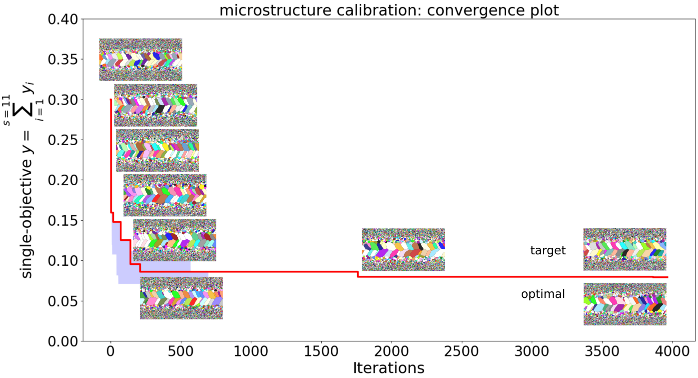


Figure 66: Reverse engineering an AM specimen through kinetic Monte Carlo (Sandia/SPPARKS). Tran et al. 2020a.

DFT + ML-IAP MD: multi-fidelity

(joint work w/ Julien Tranchida, Aidan Thompson, Tim Wildey)

Tran et al. 2020c: coupling DFT and MD. Multi-fidelity for multi-scale ICME.

Figure 67: Iteration 4: 2 LF + 2 HF

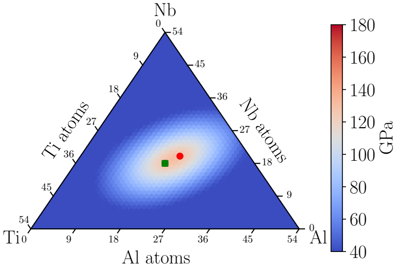
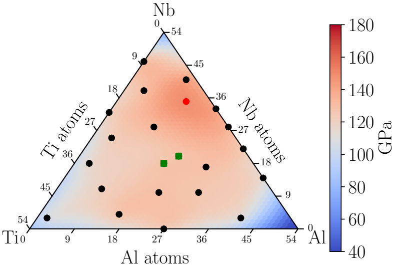


Figure 68: Iteration 24: 21 LF + 3 HF



DFT + ML-IAP MD: multi-fidelity

(joint work w/ Julien Tranchida, Aidan Thompson, Tim Wildey)

Tran et al. 2020c: coupling DFT and MD. Multi-fidelity for multi-scale ICME.

Figure 69: Iteration 35: 31 LF + 4 HF

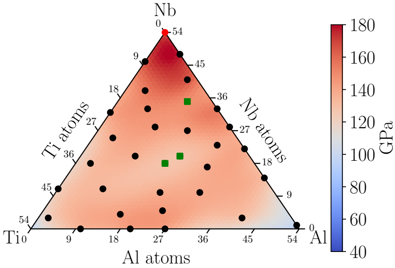
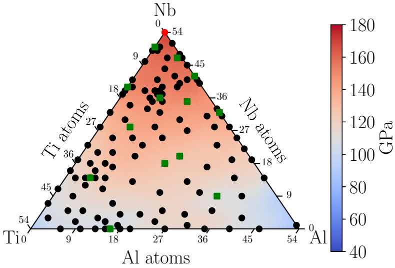


Figure 70: Iteration 130: 116 LF + 14 HF



Data-consistent for structure-property (joint work w/ Tim Wildey)

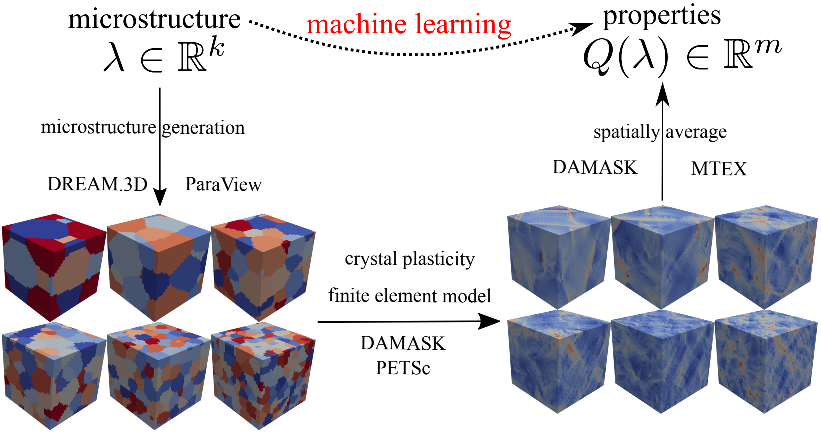


Figure 71: Microstructure-homogenized materials properties map over an ensemble of microstructures with a heteroscedastic GP.

Data-consistent for structure-property (joint work w/ Tim Wildey)

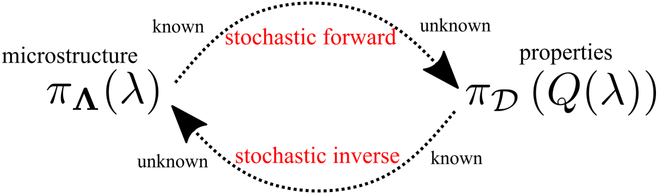


Figure 72: Stochastic forward vs. stochastic inverse problems in structure-property context.

Data-consistent for structure-property (joint work w/ Tim Wildey)

Figure 73: Ensemble average yield stress via Monte Carlo with different grain sizes

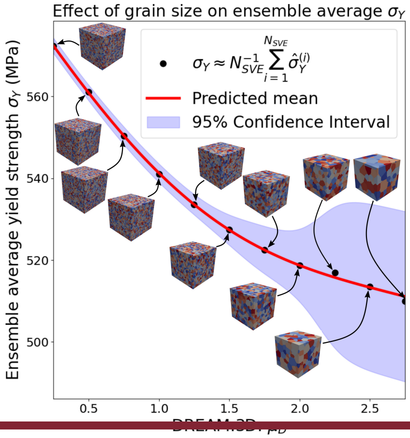
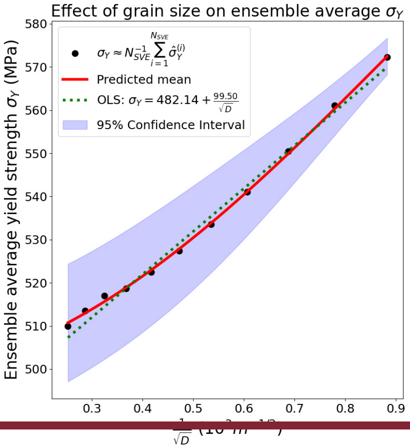


Figure 74: Comparison: GPR (ML) with uncertainty and the Hall-Petch (ordinary least square)



Data-consistent for structure-property (joint work w/ Tim Wildey)

Figure 75: Initial density and updated density: normal case

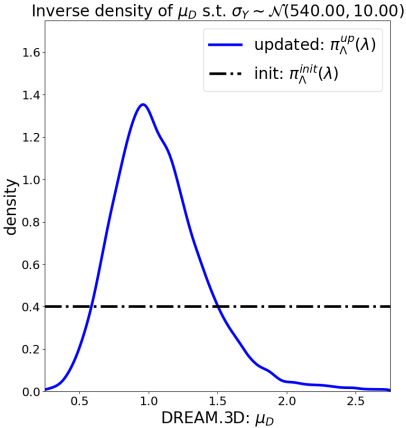
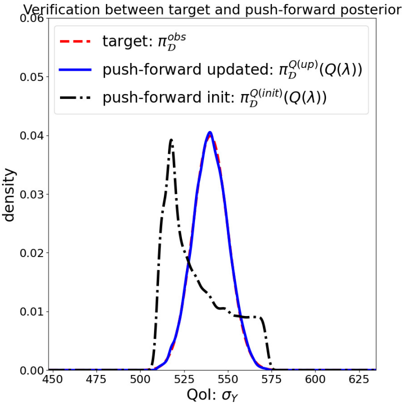


Figure 76: Comparison: Distributions of materials properties



Conclusion

Sandia National Laboratories

This talk: two parts

- theoretical / computation aspects of GP/BO
  - how to modify GP to suit the problem needs
  - how to improve BO in HPC, constrained, multi-{objective,fidelity}
  - numerical toy functions demonstration
  - plenty of open problems in Big Data, high-dimensional problems
  - state-space models
- Multi-scale engineering applications
  - density functional theory
  - molecular dynamics
  - kinetic Monte Carlo
  - computational fluid dynamics
  - ~~phase-field~~
  - crystal plasticity finite element

Thank you for listening.

Methodology:

- Anh Tran (Aug. 2021). "Scalable^3-BO: Big Data meets HPC - A scalable asynchronous parallel high-dimensional Bayesian optimization framework on supercomputers". In: Proceedings of the ASME 2021 IDETC/CIE. vol. Volume 1: 41th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers
- Anh Tran et al. (Aug. 2020d). "srMO-BO-3GP: A sequential regularized multi-objective constrained Bayesian optimization for design applications". In: Proceedings of the ASME 2020 IDETC/CIE. vol. Volume 1: 40th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers
- Anh Tran et al. (2020b). "aphBO-2GP-3B: A budgeted asynchronous-parallel multi-acquisition for constrained Bayesian optimization on high-performing computing architecture". In: arXiv preprint arXiv:2003.09436
- Anh Tran, Tim Wildey, and Scott McCann (2020). "sMF-BO-2CoGP: A sequential multi-fidelity constrained Bayesian optimization for design applications". In: Journal of Computing and Information Science in Engineering 20.3, pp. 1–15
- Anh Tran, Tim Wildey, and Scott McCann (Aug. 2019). "sBF-BO-2CoGP: A sequential bi-fidelity constrained Bayesian optimization for design applications". In: Proceedings of the ASME 2019 IDETC/CIE. vol. Volume 1: 39th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. V001T02A073. American Society of Mechanical Engineers
- Anh Tran, Minh Tran, and Yan Wang (2019). "Constrained mixed-integer Gaussian mixture Bayesian optimization and its applications in designing fractal and auxetic metamaterials". In: Structural and Multidisciplinary Optimization, pp. 1–24
- Anh Tran et al. (2019a). "pBO-2GP-3B: A batch parallel known/unknown constrained Bayesian optimization with feasibility classification and its applications in computational fluid dynamics". In: Computer Methods in Applied Mechanics and Engineering 347, pp. 827–852

Sandia National Laboratories

Applications:

- Anh Tran and Tim Wildey (2020). "Solving stochastic inverse problems for property-structure linkages using data-consistent inversion and machine learning". In: JOM 73, pp. 72–89
- Anh Tran et al. (2020c). "Multi-fidelity machine-learning with uncertainty quantification and Bayesian optimization for materials design: Application to ternary random alloys". In: The Journal of Chemical Physics 153 (7), p. 074705
- Anh Tran et al. (2020a). "An active-learning high-throughput microstructure calibration framework for process-structure linkage in materials informatics". In: Acta Materialia 194, pp. 80–92
- Stefano Travaglino et al. (2020). "Computational optimization study of transcatheter aortic valve leaflet design using porcine and bovine leaflets". In: Journal of Biomechanical Engineering 142 (1)
- Anh Tran et al. (2019b). "WearGP: A computationally efficient machine learning framework for local erosive wear predictions via nodal Gaussian processes". In: Wear 422, pp. 9–26
- Anh Tran, Lijuan He, and Yan Wang (2018). "An efficient first-principles saddle point searching method based on distributed kriging metamodels". In: ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering 4.1, p. 011006

References I

- Azimi, Javad, Alan Fern, and Xiaoli Z Fern (2010). "Batch Bayesian optimization via simulation matching". In: Advances in Neural Information Processing Systems, pp. 109–117.
- Beume, Nicola et al. (2009). "On the complexity of computing the hypervolume indicator". In: IEEE Transactions on Evolutionary Computation 13.5, pp. 1075–1082.
- Chalupka, Krzysztof, Christopher KI Williams, and Iain Murray (2013). "A framework for evaluating approximation methods for Gaussian process regression". In: Journal of Machine Learning Research 14.Feb, pp. 333–350.
- Chevalier, Clément and David Ginsbourger (2013). "Fast computation of the multi-points expected improvement with applications in batch selection". In: International Conference on Learning and Intelligent Optimization. Springer, pp. 59–69.

References II

Cho, Yigil et al. (2014). "Engineering the shape and structure of materials by fractal cut". In: Proceedings of the National Academy of Sciences 111.49, pp. 17390–17395.

Contal, Emile et al. (2013). "Parallel Gaussian process optimization with upper confidence bound and pure exploration". In: Joint European Conference on Machine Learning and Knowledge Discovery i Springer, pp. 225–240.

Couckuyt, I, T Dhaene, and P Demeester (2013). "ooDACE toolbox, A Matlab Kriging toolbox: Getting started". In: Universiteit Gent, pp. 3–15.

Couckuyt, Ivo, Tom Dhaene, and Piet Demeester (2014). "ooDACE toolbox: a flexible object-oriented Kriging implementation". In: The Journal of Machine Learning Research 15.1, pp. 3183–3186.

Couckuyt, Ivo et al. (2012). "Blind Kriging: Implementation and performance analysis". In: Advances in Engineering Software 49, pp. 1–13.

References III

Desautels, Thomas, Andreas Krause, and Joel W Burdick (2014). "Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization". In: The Journal of Machine Learning Research 15.1, pp. 3873–3923.

Digabel, Sébastien Le and Stefan M Wild (2015). "A Taxonomy of Constraints in Simulation-Based Optimization". In: arXiv preprint arXiv:1505.07881.

Forrester, Alexander IJ, András Sóbester, and Andy J Keane (2007). "Multi-fidelity optimization via surrogate modelling". In: Proceedings of the Royal Society of London A: mathematical, physical and e 463.2088, pp. 3251–3269.

Frigola, Roger, Yutian Chen, and Carl Edward Rasmussen (2014). "Variational Gaussian Process State-Space Models". In: Advances in Neural Information Processing Systems. Vol. 27. Curran Associates, Inc.

References IV

González, Javier et al. (2016). "Batch Bayesian optimization via local penalization". In: Proceedings of the 19th International Conference on Artificial Intelligence ar pp. 648–657.

Hoffman, Matthew, Eric Brochu, and Nando de Freitas (2011). "Portfolio Allocation for Bayesian Optimization". In: Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial I UAI'11. Barcelona, Spain: AUAI Press, pp. 327–336. ISBN: 9780974903972.

Huang, Deng et al. (2006). "Global optimization of stochastic black-box systems via sequential kriging meta-models". In: Journal of Global Optimization 34.3, pp. 441–466.

Kandasamy, Kirthevasan et al. (2017). "Asynchronous parallel Bayesian optimisation via Thompson sampling". In: arXiv preprint arXiv:1705.09236.

References V

- Kathuria, Tarun, Amit Deshpande, and Pushmeet Kohli (2016). "Batched Gaussian process bandit optimization via determinantal point processes". In: Advances in Neural Information Processing Systems, pp. 4206–4214.
- Kennedy, Marc C and Anthony O'Hagan (2000). "Predicting the output from a complex computer code when fast approximations are available". In: Biometrika 87.1, pp. 1–13.
- Lawrence, Neil D (2016). "Introduction to gaussian processes". In: MLSS 8, p. 504. URL: inverseprobability.com/talks/slides/gp_mlss16.pdf.
- Le Gratiet, Loic and Josselin Garnier (2014). "Recursive co-kriging model for design of computer experiments with multiple levels of fidelity". In: International Journal for Uncertainty Quantification 4.5.
- Li, Mu, James Tin-Yau Kwok, and Baoliang Lü (2010). "Making large-scale Nyström approximation possible". In: ICML 2010-Proceedings, 27th International Conference on Machine Learning p. 631.

References VI

- Mahoney, Michael W (2016). "Lecture notes on randomized linear algebra". In: arXiv preprint arXiv:1608.04481.
- Marmin, Sébastien, Clément Chevalier, and David Ginsbourger (2016). "Efficient batch-sequential Bayesian optimization with moments of truncated Gaussian vectors". In: arXiv preprint arXiv:1609.02700.
- Martinsson, Per-Gunnar and Joel A Tropp (2020). "Randomized numerical linear algebra: Foundations and algorithms". In: Acta Numerica 29, pp. 403–572.
- Mockus, Jonas (1975). "On Bayesian methods for seeking the extremum". In: Optimization Techniques IFIP Technical Conference. Springer, pp. 400–404.
- — (1982). "The Bayesian approach to global optimization". In: System Modeling and Optimization, pp. 473–481.
- Perdikaris, P et al. (2015). "Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random fields". In: Proc. R. Soc. A. Vol. 471. 2179. The Royal Society, p. 20150018.

References VII

- Perdikaris, Paris and George Em Karniadakis (2016). "Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond". In: Journal of The Royal Society Interface 13.118, p. 20151107.
- Quiñonero-Candela, Joaquin and Lars Kai Hansen (2004). "Learning with uncertainty-Gaussian processes and relevance vector machines". In: Technical University of Denmark, Copenhagen.
- Quiñonero-Candela, Joaquin and Carl Edward Rasmussen (2005). "A unifying view of sparse approximate Gaussian process regression". In: Journal of Machine Learning Research 6.Dec, pp. 1939–1959.
- Quiñonero-Candela, Joaquin, Carl Edward Rasmussen, and Christopher KI Williams (2007). "Approximation methods for Gaussian process regression". In: Large-scale kernel machines, pp. 203–224.
- Rasmussen, Carl Edward (2006). Gaussian processes in machine learning. MIT Press.

References VIII

- Scott, Warren, Peter Frazier, and Warren Powell (2011). "The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression". In: SIAM Journal on Optimization 21.3, pp. 996–1026.
- Shah, Amar and Zoubin Ghahramani (2015). "Parallel predictive entropy search for batch global optimization of expensive objective functions". In: Advances in Neural Information Processing Systems, pp. 3330–3338.
- Solnik, Benjamin et al. (2017). "Bayesian Optimization for a Better Dessert". In.
- Srinivas, Niranjan et al. (2009). "Gaussian process optimization in the bandit setting: No regret and experimental design". In: arXiv preprint arXiv:0912.3995.
- Srinivas, Niranjan et al. (2012). "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting". In: IEEE Transactions on Information Theory 58.5, pp. 3250–3265.

References IX

- Tran, Anh (Aug. 2021). "Scalable^3-BO: Big Data meets HPC - A scalable asynchronous parallel high-dimensional Bayesian optimization framework on supercomputers". In: Proceedings of the ASME 2021 IDETC/CIE. Vol. Volume 1: 41th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers.
- Tran, Anh, Lijuan He, and Yan Wang (2018). "An efficient first-principles saddle point searching method based on distributed kriging metamodels". In: ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B 4.1, p. 011006.
- Tran, Anh, Minh Tran, and Yan Wang (2019). "Constrained mixed-integer Gaussian mixture Bayesian optimization and its applications in designing fractal and auxetic metamaterials". In: Structural and Multidisciplinary Optimization, pp. 1–24.

References X

Tran, Anh and Tim Wildey (2020). "Solving stochastic inverse problems for property-structure linkages using data-consistent inversion and machine learning". In: JOM 73, pp. 72–89.

Tran, Anh, Tim Wildey, and Scott McCann (Aug. 2019). "sBF-BO-2CoGP: A sequential bi-fidelity constrained Bayesian optimization for design applications". In: Proceedings of the ASME 2019 IDETC/CIE. Vol. Volume 1: 39th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. V001T02A073. American Society of Mechanical Engineers.

— (2020). "sMF-BO-2CoGP: A sequential multi-fidelity constrained Bayesian optimization for design applications". In: Journal of Computing and Information Science in Engineering 20.3, pp. 1–15.

References XI

- Tran, Anh et al. (2019a). "pBO-2GP-3B: A batch parallel known/unknown constrained Bayesian optimization with feasibility classification and its applications in computational fluid dynamics". In: Computer Methods in Applied Mechanics and Engineering 347, pp. 827–852.
- Tran, Anh et al. (2019b). "WearGP: A computationally efficient machine learning framework for local erosive wear predictions via nodal Gaussian processes". In: Wear 422, pp. 9–26.
- Tran, Anh et al. (2020a). "An active-learning high-throughput microstructure calibration framework for process-structure linkage in materials informatics". In: Acta Materialia 194, pp. 80–92.
- Tran, Anh et al. (2020b). "aphBO-2GP-3B: A budgeted asynchronous-parallel multi-acquisition for constrained Bayesian optimization on high-performing computing architecture". In: arXiv preprint arXiv:2003.09436.

References XII

- Tran, Anh et al. (2020c). "Multi-fidelity machine-learning with uncertainty quantification and Bayesian optimization for materials design: Application to ternary random alloys". In: The Journal of Chemical Physics 153 (7), p. 074705.
- Tran, Anh et al. (Aug. 2020d). "srMO-BO-3GP: A sequential regularized multi-objective constrained Bayesian optimization for design applications". In: Proceedings of the ASME 2020 IDETC/CIE. Vol. Volume 1: 40th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers.
- Travaglino, Stefano et al. (2020). "Computational optimization study of transcatheter aortic valve leaflet design using porcine and bovine leaflets". In: Journal of Biomechanical Engineering 142 (1).
- Vanhatalo, Jarno et al. (2012). "Bayesian modeling with Gaussian processes using the GPstuff toolbox". In: arXiv preprint arXiv:1206.5754.

References XIII

- Vanhatalo, Jarno et al. (2013). "GPstuff: Bayesian modeling with Gaussian processes". In: Journal of Machine Learning Research 14.Apr, pp. 1175–1179.
- Wang, Ziyu et al. (2013). "Bayesian optimization in high dimensions via random embeddings". In: AAAI Press/International Joint Conferences on Artificial Intelligence.
- Wang, Ziyu et al. (2016). "Bayesian optimization in a billion dimensions via random embeddings". In: Journal of Artificial Intelligence Research 55, pp. 361–387.
- Williams, Christopher and Matthias Seeger (2001). "Using the Nyström method to speed up kernel machines". In: Advances in neural information processing systems 13, pp. 682–688.
- Wu, Jian and Peter Frazier (2016). "The parallel knowledge gradient method for batch Bayesian optimization". In: Advances in Neural Information Processing Systems, pp. 3126–3134.

References XIV

Xiao, Manyu et al. (2018). "Extended Co-Kriging interpolation method based on multi-fidelity data". In: Applied Mathematics and Computation 323, pp. 120–131.