Multi-objective optimization of an agent-based simulator

Mickaël Binois (Inria Sophia Antipolis - Méditerranée)
mickael.binois@inria.fr

joint work with Nicholson Collier (Argonne National Laboratory), Mert Edali (University of Chicago) & Jonathan Ozik (Argonne National Laboratory, University of Chicago)

MASCOT-NUM Workshop on Stochastic Simulators, online

March 11, 2020

Context

Let us consider an expensive-to-evaluate **black-box** simulator:

$$f : \mathcal{X} \subset \mathbb{R}^d \rightarrow G \subset \mathbb{R}^m$$

Typically :

- $1 \leq d \leq 50$
- $1 \leq m \leq 5$
- one evaluation may take minutes up to days
- evaluations may be noisy, with variance depending on $\mathbf{x}$

Aim: finding best compromise solutions between objectives with limited evaluations

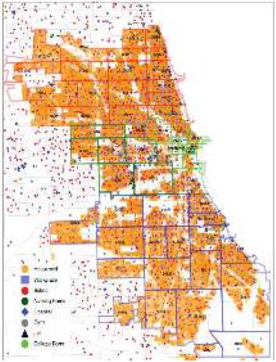
Outline

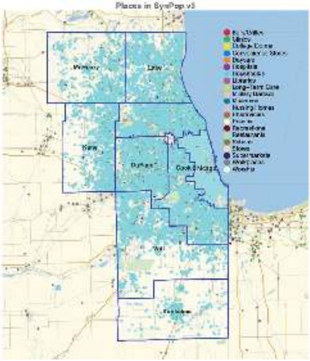
Motivating problem: Detailed Agent-based modeling

ChiSIM is an agent-based simulation model of the urban area of Chicago.

It includes a virtual population of 2.7 million residents in terms of people (behaviors and social interations), places (1.2 million unique geolocations) and hourly activity schedules.

C. M. Macal, N. T. Collier and J. Ozik, E. R. Tatara, and J. T. Murphy.
chiSIM: An agent-based simulation model of social interactions in a large urban area.
Winter Simulation Conference, 2018

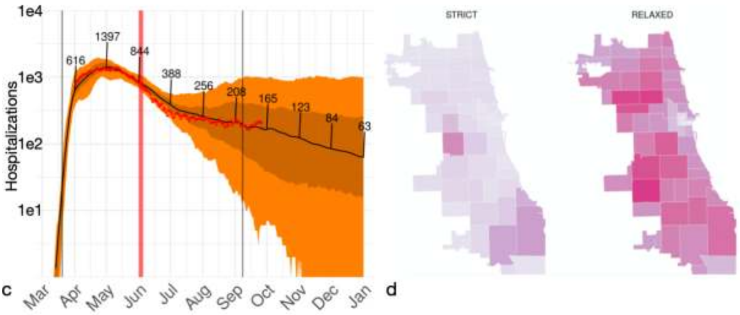




CityCovid: COVID-19 Modeling and Learning

Amid the COVID-19 crisis, chiSIM is used to study the epidemic outbreak and support Public Health departments.

J. Ozik, J. Wozniak, N. Collier, C. Macal and M. Binois.
A Population Data-Driven Workflow for COVID-19 Modeling and Learning.
International Journal of High Performance Computing Applications (accepted), 2020+

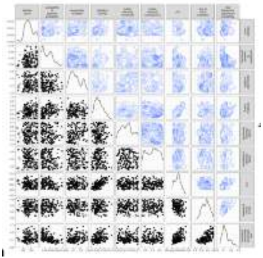


Calibrating the CityCOVID model

Empirical data from hospital systems and public health departments are used to parameterize CityCOVID and provide targets for calibration (in particular, cumulative deaths and point prevalence hospitalization).

Remaining parameters (9) to calibrate include: out-of-household activity levels, person-to-person transmission reduction, ...

Sequential incremental mixture approximate Bayesian computation is used [Rutter et al., 2019]:



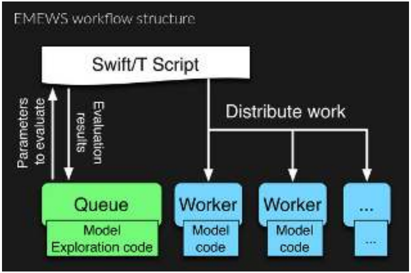
Still, a *best* set of calibration results is important for scenarii assessment.

CityCOVID: computational details

CityCOVID is run based on the REPAST agent-based modeling suite `https://repast.github.io`

EMEWS is used to handle the whole model exploration workflow `http://emews.org`

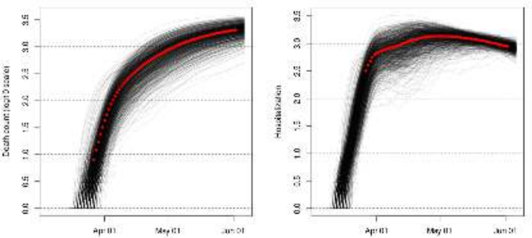
J. Ozik, N. Collier, J. M. Wozniak and C. Spagnuolo.
From Desktop to Large-Scale Model Exploration with Swift/T.
Winter Simulation Conference, 2016.



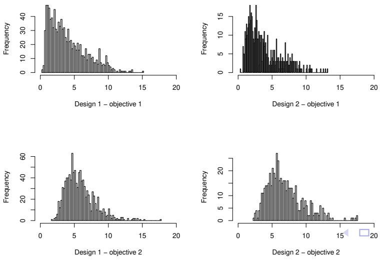
It runs on Argonne's ALCF Theta supercomputer (4932 nodes of 64 cores, 11.7 petaflops total).

Towards noisy multi-objective optimization

Outputs of 1000 replicates of the same parameters: (observed)



(weighted) aggregated errors with respect to real data:



CityCOVID challenges for multi-objective optimization

1) Signal-to-noise ratio is very low, larger evaluation budgets are necessary – but the total budget is still limited.

2) Noise is heteroscedastic (and non-Gaussian).

3) Hundreds of runs can (must) be conducted in parallel.

4) The simulation model is moderately fast, approximately 10 minutes.

5) So must be the model exploration part, with focus on cluster usage.

We show methods to cope with these features with Bayesian optimization in the following.

Outline

Concepts in Multi-objective Optimization (MOO)

A solution minimizing every objective at once usually does not exist.

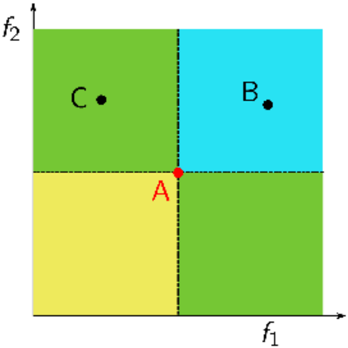
Pareto dominance

Vector A dominates vector B if:

- $\forall i \in \{1, \ldots, n\}, a_i \leq b_i$
- $\exists j \in \{1, \ldots, n\}, a_j < b_j$

Pareto optimality

A is Pareto optimal if it is non-dominated.



- **Pareto set** (PS): set of all optimal points in the variable space
- **Pareto front** (PF): image of the Pareto set in the objective space

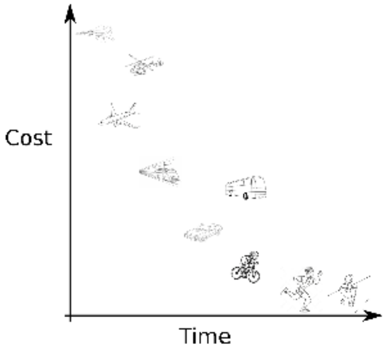
Concepts in Multi-objective Optimization (MOO)

A solution minimizing every objective at once usually does not exist.

Pareto dominance

Vector A dominates vector B if:

- $\forall i \in \{1, \ldots, n\}$, $a_i \leq b_i$
- $\exists j \in \{1, \ldots, n\}, a_j < b_j$



Pareto optimality

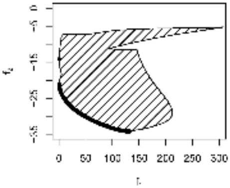
A is Pareto optimal if it is non-dominated.

- **Pareto set** (PS): set of all optimal points in the variable space
- **Pareto front** (PF): image of the Pareto set in the objective space

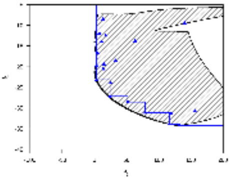
Noisy MOO

How to define a reference when the true Pareto front is unknown?

Deterministic setup (no noise):



Ground truth

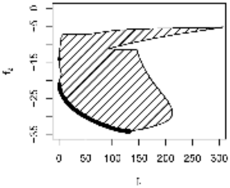


Possible result

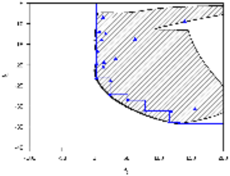
Noisy MOO

How to define a reference when the true Pareto front is unknown?

Deterministic setup (no noise):

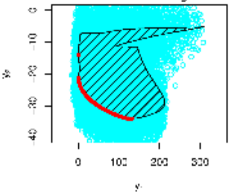


Ground truth

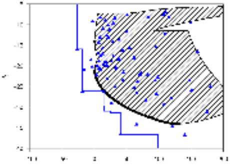


Possible result

Noisy objectives setup:



Ground truth



Possible result

Problem definition

The deterministic MOO problem is:

$$\text{find } \mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} \left(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\right)$$

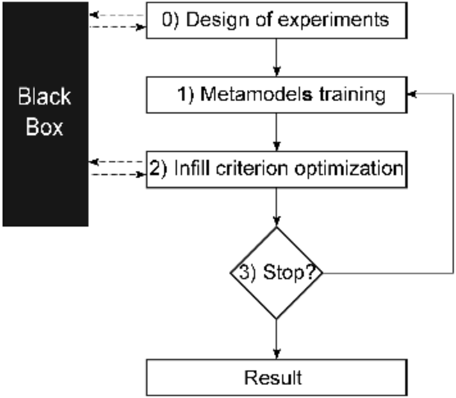
The noisy MOO problem is often formulated as (see e.g., [Hunter et al., 2019]):

$$\text{find } \mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} \left(\mathbb{E}[f_1(\mathbf{x})], \ldots, \mathbb{E}[f_m(\mathbf{x})]\right)$$

Other definitions are possible:

- to take into account reliability, such as the Pareto optimal probability, see, e.g., [Rivier and Congedo, 2018];
- set-based (or joint) approach, e.g., considering the Pareto front of a Gaussian process, see, e.g., [Binois et al., 2015].

General solving procedure in MO Bayesian Optimization



For example:

0. Maximin Latin Hypercubes Samples
1. Gaussian process models
2. **Expected Hypervolume Improvement**
3. Budget

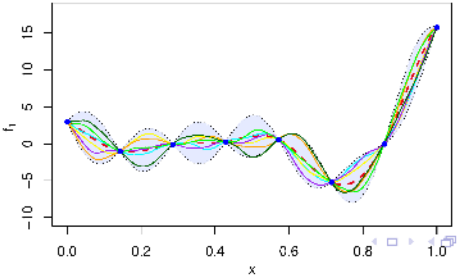
Surrogate modeling: Gaussian process regression

We use a zero mean GP prior on f, with covariance k: $Y \sim \mathcal{GP}(0, k)$.

MVN conditional identities give directly the result on $(\mathbf{x}_i, f(\mathbf{x}_i))_{1 \leq i \leq n}$:

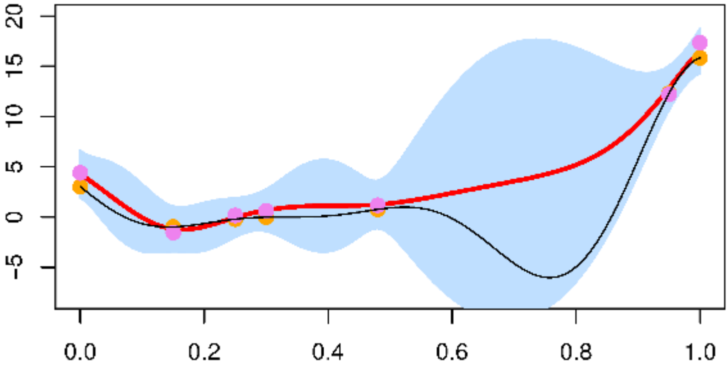
$$
\begin{aligned}
&Y(\mathbf{x})|\mathbf{y} \sim \mathcal{GP}(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x})) \text{ with} \\
\mu_n(\mathbf{x}) &= \mathbb{E}(Y(\mathbf{x})|\mathbf{y}) = \mathbf{k}(\mathbf{x})^\top \mathbf{K}_n^{-1}\mathbf{y}, \\
\sigma_n^2(\mathbf{x}) &= \mathbb{V}\mathrm{ar}(Y(\mathbf{x})|\mathbf{y}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\top \mathbf{K}_n^{-1}\mathbf{k}(\mathbf{x}), \text{ where}
\end{aligned}
$$

$\mathbf{y} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$, $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\top$ and $\mathbf{K}_n = (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i,j \leq n}$.



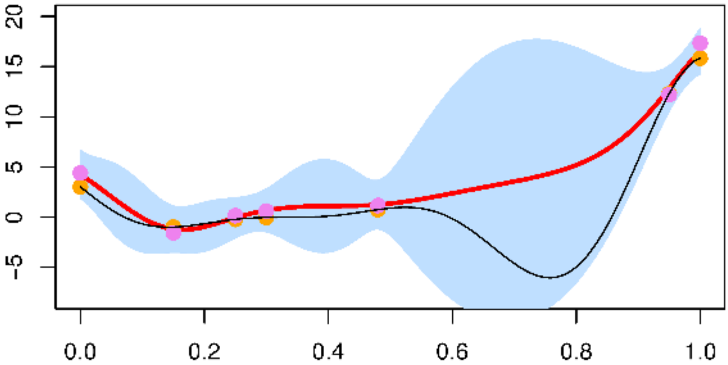
1) Noisy observations

GPs readily handle Gaussian noise, e.g., through the estimation of a constant noise term.



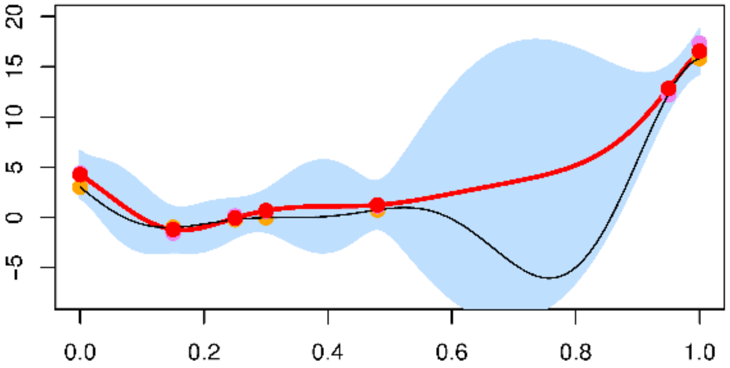
1) Noisy observations

Sometimes it is desirable to filter out the noise, one such technique is reinterpolation (e.g., [Forrester and Keane, 2009, Picheny and Ginsbourger, 2014]).



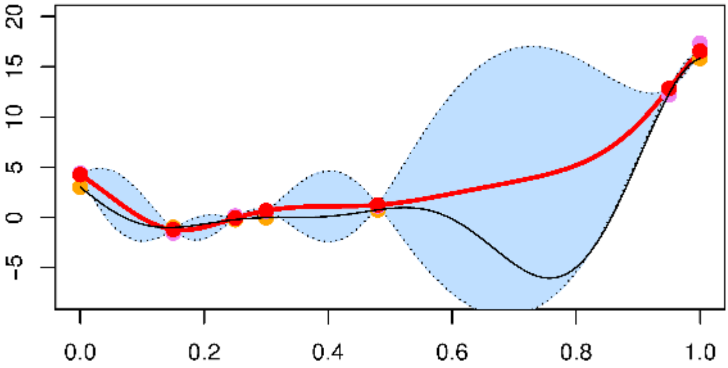
1) Noisy observations

Sometimes it is desirable to filter out the noise, one such technique is reinterpolation (e.g., [Forrester and Keane, 2009, Picheny and Ginsbourger, 2014]).



1) Noisy observations

Sometimes it is desirable to filter out the noise, one such technique is reinterpolation (e.g., [Forrester and Keane, 2009, Picheny and Ginsbourger, 2014]).

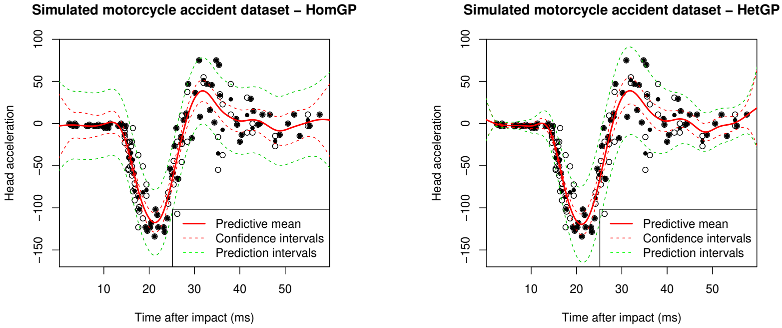


Heteroskedastic Gaussian process modeling

The noise variance may be input-dependent.

Observation model: $y(\mathbf{x}_i) = \bar{f}(\mathbf{x}_i) + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, r(\mathbf{x}_i))$.

We may be interested in $P(\hat{f}|\mathcal{A}_n)$ but also in $P(y|\mathcal{A}_n)$; $\mathcal{A}_n = (\mathbf{x}_i, f(\mathbf{x}_i))$.



Several methods for GPs, with a second GP on the log-noise variance, e.g., [Goldberg et al., 1997, Kersting et al., 2007, Ankenman et al., 2008, Lázaro-Gredilla and Titsias, 2011, Binois et al., 2018]

Handling large data

Standard GP struggles when n reaches a few thousands, due to the cubic computational cost of matrix inversion.

Recent implementations (e.g., GPyTorch [Gardner et al., 2018], GPflow [Matthews et al., 2017]) leverage GPU implementation, advanced linear algebra or inducing points to scale to large n.

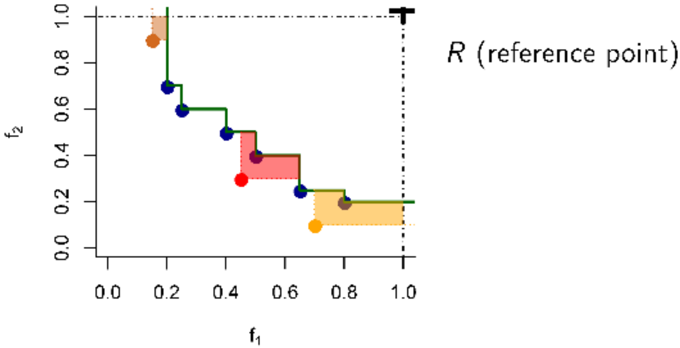
Other ideas include the use of covariance functions with compact support leading to sparse covariance matrices.

A simple yet powerful alternative is to rely on local GPs, defined on a subset of the large DoE:

- handle non-stationarity,
- tunable cost (via the size of the subset),
- adapted to the prediction location(s), via nearest neighbors (or GP design criteria),
- highly parallel.

2) MO infill criterion - Hypervolume Improvement

One standard MO metric is the Hypervolume Improvement $I_{\mathcal{H}}$, i.e., the volume added to the current Pareto front by a new observation.

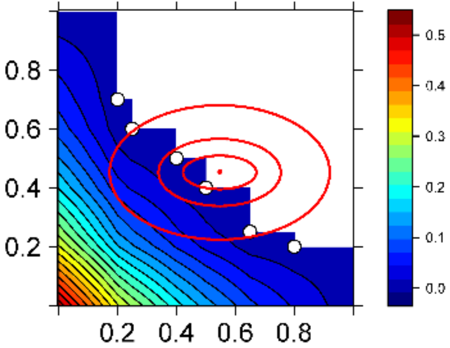


The corresponding generalization of EI is the Expected Hypervolume Improvement [Emmerich et al., 2011]: $EHI(\mathbf{x}) = \mathbb{E}(I_{\mathcal{H}}(\mathbf{Y}(\mathbf{x}))|\mathcal{A}_n)$.

EHI has a closed-form for 2 or 3 objectives, or can be computed by sampling from the GP posterior at $\mathbf{x}$.

2) MO infill criterion - Hypervolume Improvement

One standard MO metric is the Hypervolume Improvement $I_{\mathcal{H}}$, i.e., the volume added to the current Pareto front by a new observation.



The corresponding generalization of EI is the Expected Hypervolume Improvement [Emmerich et al., 2011]: $EHI(\mathbf{x}) = \mathbb{E}(I_{\mathcal{H}}(\mathbf{Y}(\mathbf{x}))|\mathcal{A}_n)$.

EHI has a closed-form for 2 or 3 objectives, or can be computed by sampling from the GP posterior at $\mathbf{x}$.

Other MOO acquisition function

Hypervolume-based methods are popular, see e.g., [Feliot et al., 2017, Zhang and Golovin, 2020].

But other multi-objective performance indicators (e.g., [Audet et al., 2020] can be used to define an improvement (e.g., [Svenson and Santner, 2016]).

Other criteria include:

- entropy based criteria, e.g., [Hernández-Lobato et al., 2016];
- step-wise uncertainty reduction criteria, e.g., [Picheny, 2015, Binois, 2015];
- scalarization/aggregation functions, see e.g., [Knowles, 2006, Rojas-Gonzalez and Van Nieuwenhuyse, 2020, Emmerich et al., 2020]...

Here we choose the hypervolume metric.

Batching

A simple and efficient way of handling parallel evalutation is by selecting q different reference points [Parr, 2012, Gaudrie et al., 2020].

The use of kriging believer/constant liar heuristics [Ginsbourger et al., 2010] is also possible (a.k.a. hallucination).

[Daulton et al., 2020] proposed a batch (differentiable) expected hypervolume improvement for deterministic problems, relying on the reparameterization trick [Wilson et al., 2017] (writing the acquisition function in integral form). Then Monte Carlo estimation is used, relying on the Sample Average Approximation (fixed random seed).

None is really adapted to massively parallel systems.

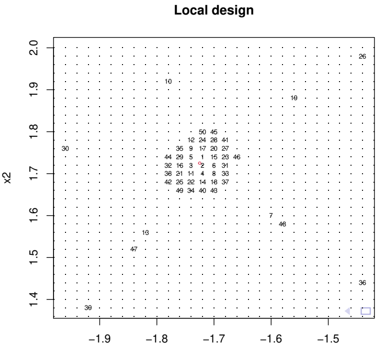
Outline

Choice of GP modeling

Given the low signal-to-noise ratio, we enforce a minimal degree of replication of 10.

For speed and parallelization, we use local GPs:

- `laGP` is used to select local designs [Gramacy, 2016];
- `hetGP` is used for heteroskedastic GP modeling at these selected designs [Binois and Gramacy, 2020];
- `GPareto` is used for EHI computation [Binois and Picheny, 2019].



Expected Hypervolume improvement optimization

Ultimately, only evaluated designs are considered for outputs (i.e., not a prediction).

Still, for hypervolume improvement computations, we use the predicted means on the evaluated designs to obtain a current Pareto front (and filter out noise).

For optimization, we start by evaluation of EHI on a large random discrete set.

Then, we perform local optimizations.

In the end, we have a set of p designs with large EHI values.

Handling large batches and favoring replication

We have to allocate up to $q = 4000$ evaluations per iteration.

Optimizing a batch-EHI is out of reach ($q \times d = 4000 \times 9$ variables).

Instead:

- we allocate part of the budget to refine the currently identified non-dominated points.
- the rest is allocated based on a portfolio allocation scheme.
- it enforces replication, which is beneficial for both estimating the variance and fast GP inference.

Portfolio allocation for batch selection (1)

We follow the idea by [Guerreiro et al.,2016] of using the Sharpe ratio to select a population in multi-objective evolutionary algorithms.

A. P. Guerreiro & C. M. Fonseca.
Hypervolume Sharpe-Ratio indicator: Formalization and first theoretical results
International Conference on Parallel Problem Solving from Nature, 814-823, 2016.

Sharpe-Ratio Maximization (as in [Guerreiro et al.,2016])

Let $A = \left\{a^{(1)}, \ldots, a^{(p)}\right\}$ be a non-empty set of assets, let vector $\mathbf{r} \in \mathbb{R}^p$ denote the expected return of these assets and matrix $Q \in \mathbb{R}^{p \times p}$ denote the return covariance between pairs of assets. Let $\mathbf{z} \in [0,1]^p$ be the investment vector where z_i denotes the investment in asset $a^{(i)}$. The Sharpe-Ratio maximization problem is defined as

$$\max_{\mathbf{z} \in [0,1]^p} h(\mathbf{z}) = \frac{\mathbf{r}^\top \mathbf{z} - r_f}{\sqrt{\mathbf{z}^\top Q \mathbf{z}}} s.t. \sum_{i=1}^{n} z_i = 1$$

and r_f the return of riskless asset and h the Sharpe ratio.

Portfolio allocation for batch selection (2)

Solving the Sharpe-Ratio maximization can be solved with quadratic programming.

The possible assets here are the designs with large EHI values $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}$ that were found by sampling and multi-start optimization.

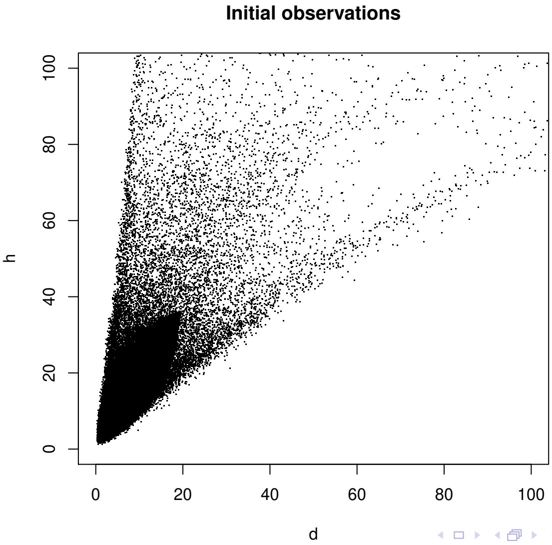
$r_f = 0$: riskless assets have zero hypervolume improvement.

To obtain the corresponding return and covariance:

1. draw L conditional GP simulations at $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}$ to get $(y_1(\mathbf{x}_i)^{(j)}, y_2(\mathbf{x}_i)^{(j)})_{1 \leq i \leq p; 1 \leq j \leq L}$
2. compute the hypervolume improvement of $(y_1(\mathbf{x}_i)^{(j)}, y_2(\mathbf{x}_i)^{(j)})_{1 \leq i \leq p; 1 \leq j \leq L}$
3. compute the corresponding return and covariance of $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}$

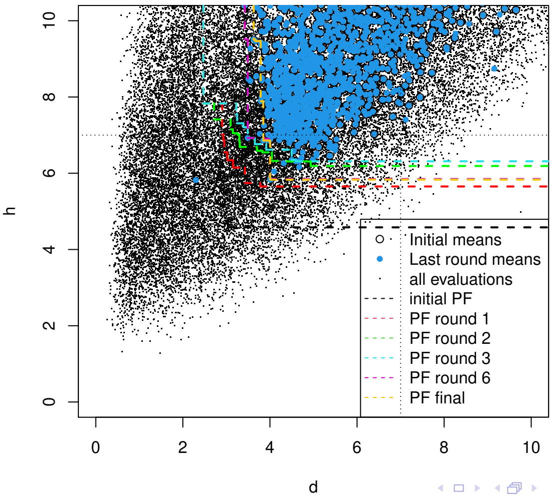
Results on CityCOVID

Initial state: 50585 observations (5075 unique designs) from IMABC



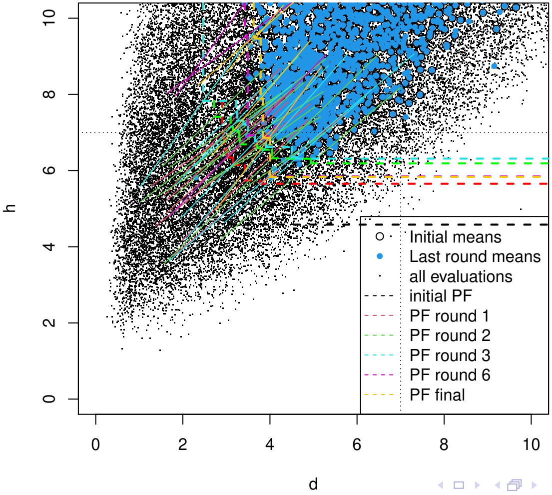
Results on CityCOVID

Final state: 86996 observations (5875 unique designs)



Results on CityCOVID

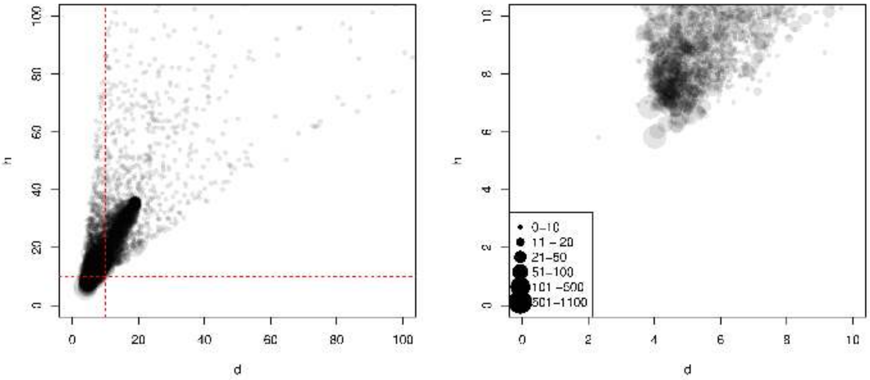
Final state: 86996 observations (5875 unique designs)



Results on CityCOVID

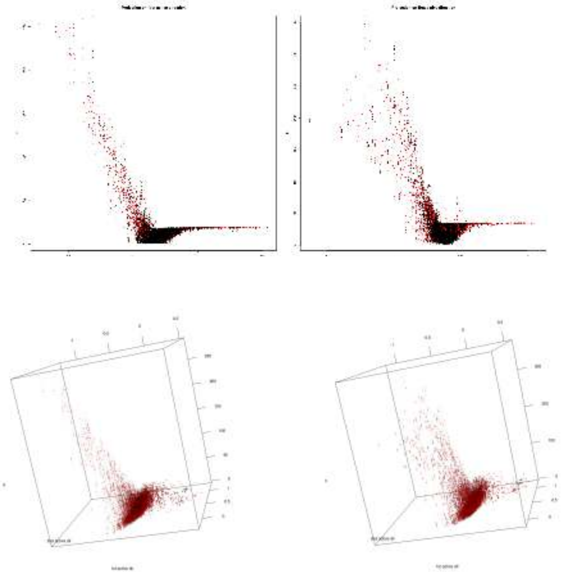
Final state: 86996 observations (5875 unique designs)

Replication map



Results on CityCOVID

Active subspace analysis, e.g., [Constantine, 2015, Wycoff et al., 2021]:



Outline

Summary and perspectives

Simulation models are challenging, here for MOO, but with many opportunities [Baker et al., 2020].

Individual evaluations may contain very little information, it implies:

- a necessity of batching and replicating;
- with an adaptive allocation;
- and fast to compute criteria.

Here we tried to have a decentralized approach to benefit from the massively parallel architecture.

Perspectives include: more accurate noise modeling, decoupled setting, increasing the number of objectives and less myopic approaches.

Ankenman, B., Nelson, B. L., and Staum, J. (2008).
Stochastic kriging for simulation metamodeling.
In *2008 Winter Simulation Conference*, pages 362–370. IEEE.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., and Salomon, L. (2020).
Performance indicators in multiobjective optimization.
European journal of operational research.

Baker, E., Barbillon, P., Fadikar, A., Gramacy, R. B., Herbei, R., Higdon, D., Huang, J., Johnson, L. R., Ma, P., Mondal, A., et al. (2020).
Analyzing stochastic computer models: A review with opportunities.
arXiv preprint arXiv:2002.01321.

References II

Binois, M. (2015).
Uncertainty quantification on Pareto fronts and high-dimensional strategies in Bayesian optimization, with applications in multi-objective automotive design.
PhD thesis, Saint-Etienne, EMSE.

Binois, M., Ginsbourger, D., and Roustant, O. (2015).
Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations.
European journal of operational research, 243(2):386–394.

Binois, M. and Gramacy, R. B. (2020).
hetGP: Heteroskedastic Gaussian Process Modeling and Design under Replication.
R package version 1.1.3.

References III

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018).
Practical heteroscedastic Gaussian process modeling for large simulation experiments.
Journal of Computational and Graphical Statistics, 27(4):808–821.

Binois, M. and Picheny, V. (2019).
GPareto: An R package for Gaussian-process-based multi-objective optimization and analysis.
Journal of Statistical Software, 89(8):1–30.

Constantine, P. G. (2015).
Active subspaces: Emerging ideas for dimension reduction in parameter studies.
SIAM.

References IV

Daulton, S., Balandat, M., and Bakshy, E. (2020).
Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization.
Advances in Neural Information Processing Systems, 33.

Emmerich, M. T., Deutz, A. H., and Klinkenberg, J. W. (2011).
Hypervolume-based expected improvement: Monotonicity properties and exact computation.
In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2147–2154. IEEE.

Emmerich, M. T., Yang, K., and Deutz, A. H. (2020).
Infill criteria for multiobjective Bayesian optimization.
In *High-Performance Simulation-Based Optimization*, pages 3–16. Springer.

References V

Feliot, P., Bect, J., and Vazquez, E. (2017).
A Bayesian approach to constrained single-and multi-objective optimization.
Journal of Global Optimization, 67(1-2):97–133.

Forrester, A. I. and Keane, A. J. (2009).
Recent advances in surrogate-based optimization.
Progress in aerospace sciences, 45(1-3):50–79.

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G. (2018).
Gpytorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration.
arXiv preprint arXiv:1809.11165.

References VI

Gaudrie, D., Le Riche, R., Picheny, V., Enaux, B., and Herbert, V. (2020).
Targeting solutions in Bayesian multi-objective optimization: sequential and batch versions.
Annals of Mathematics and Artificial Intelligence, 88(1):187–212.

Ginsbourger, D., Le Riche, R., and Carraro, L. (2010).
Kriging is well-suited to parallelize optimization.
In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Springer.

Goldberg, P. W., Williams, C. K., and Bishop, C. M. (1997).
Regression with input-dependent noise: A Gaussian process treatment.
Advances in neural information processing systems, 10:493–499.

References VII

Gramacy, R. B. (2016).
laGP: Large-scale spatial modeling via local approximate Gaussian processes in R.
Journal of Statistical Software, 72(1):1–46.

Hernández-Lobato, D., Hernandez-Lobato, J., Shah, A., and Adams, R. (2016).
Predictive entropy search for multi-objective Bayesian optimization.
In *International Conference on Machine Learning*, pages 1492–1501. PMLR.

Hunter, S. R., Applegate, E. A., Arora, V., Chong, B., Cooper, K., Rincón-Guevara, O., and Vivas-Valencia, C. (2019).
An introduction to multiobjective simulation optimization.
ACM Transactions on Modeling and Computer Simulation (TOMACS), 29(1):1–36.

Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. (2007).
Most likely heteroscedastic Gaussian process regression.
In *Proceedings of the 24th international conference on Machine learning*, pages 393–400.

Knowles, J. (2006).
Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems.
IEEE Transactions on Evolutionary Computation, 10(1):50–66.

Lázaro-Gredilla, M. and Titsias, M. K. (2011).
Variational heteroscedastic Gaussian process regression.
In *ICML*.

References IX

Matthews, A. G. d. G., Van Der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017).
Gpflow: A Gaussian process library using tensorflow.
J. Mach. Learn. Res., 18(40):1–6.

Parr, J. M. (2012).
Improvement Criteria for Constraint Handling and Multiobjective Optimization.
PhD thesis, University of Southampton.

Picheny, V. (2015).
Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction.
Statistics and Computing, 25(6):1265–1280.

References X

Picheny, V. and Ginsbourger, D. (2014).
Noisy kriging-based optimization methods: a unified implementation within the diceoptim package.
Computational Statistics & Data Analysis, 71:1035–1053.

Rivier, M. and Congedo, P. M. (2018).
Surrogate-Assisted Bounding-Box Approach Applied to Constrained Multi-Objective Optimisation Under Uncertainty.
PhD thesis, Inria Saclay Ile de France.

Rojas-Gonzalez, S. and Van Nieuwenhuyse, I. (2020).
A survey on kriging-based infill algorithms for multiobjective simulation optimization.
Computers & Operations Research, 116:104869.

References XI

Rutter, C. M., Ozik, J., DeYoreo, M., Collier, N., et al. (2019).
Microsimulation model calibration using incremental mixture approximate Bayesian computation.
Annals of Applied Statistics, 13(4):2189–2212.

Svenson, J. and Santner, T. (2016).
Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models.
Computational Statistics & Data Analysis, 94:250 – 264.

Wilson, J., Moriconi, R., Hutter, F., and Deisenroth, M. P. (2017).
The reparameterization trick for acquisition functions.
In *Proceedings of BayesOpt 2017*.

Wycoff, N., Binois, M., and Wild, S. M. (2021).
Sequential learning of active subspaces.
Journal of Computational and Graphical Statistics, pages 1–33.

Zhang, R. and Golovin, D. (2020).
Random hypervolume scalarizations for provable multi-objective black box optimization.
In *International Conference on Machine Learning*, pages 11096–11105. PMLR.

Thank you for your attention!

Questions?