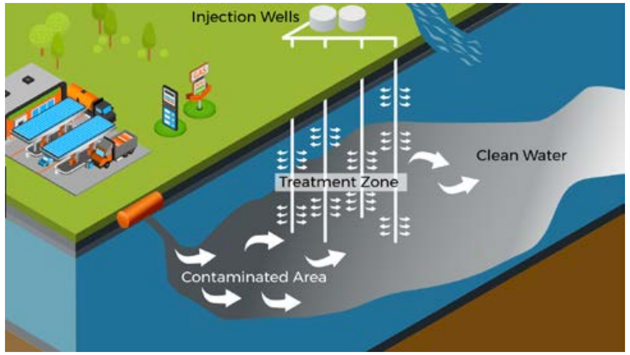
# Surrogate model guided optimization of expensive black-box multi-objective problems

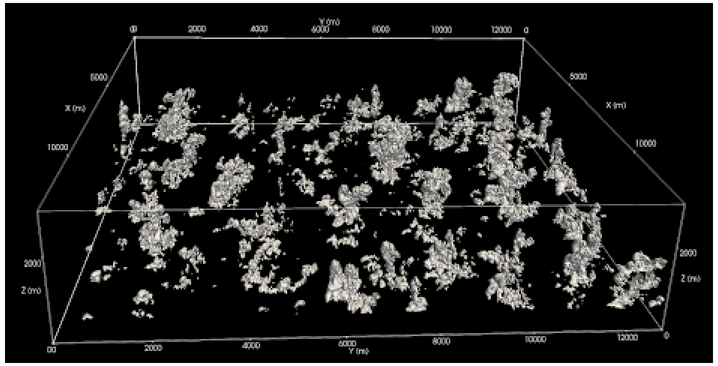
## - A posteriori methods -

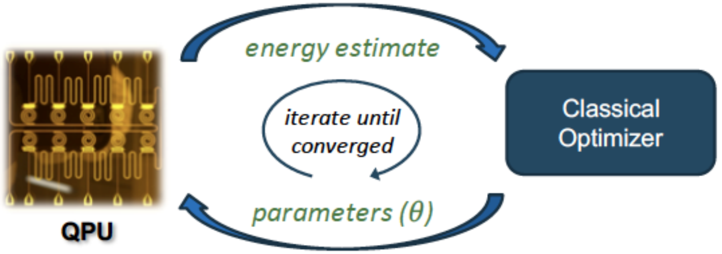
Juliane Mueller
Juliane.Mueller@nrel.gov
AI, Learning and Intelligent System Group

# Computationally expensive optimization problems arise in many science areas



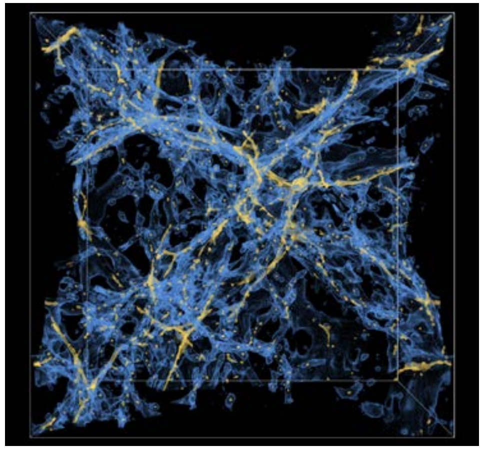
Groundwater cleanup

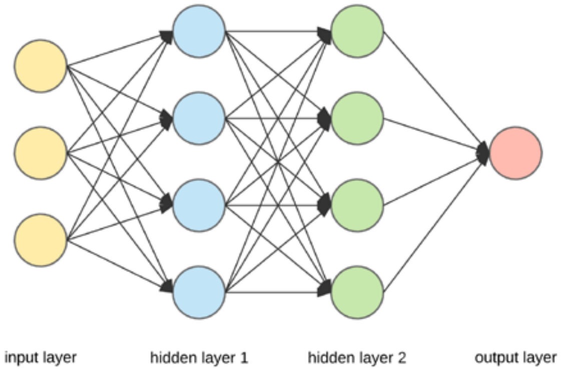

Cloud simulations



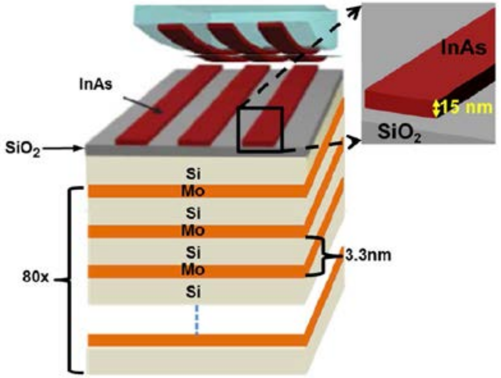
Quantum computing


Combustion


Cosmology



Deep Learning

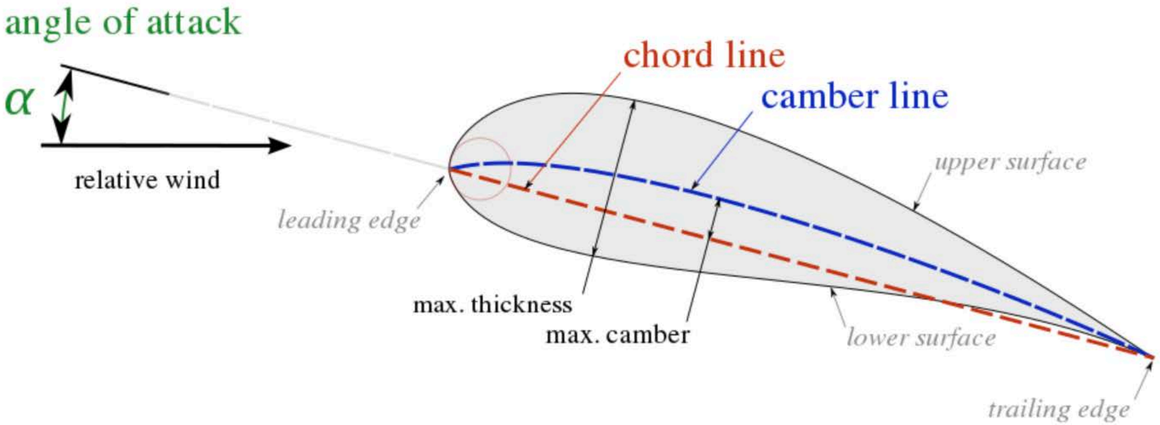


Materials science

And many many more....

# In some applications we want to optimize more than one objective function

E.g. airfoil design: maximize lift, minimize drag



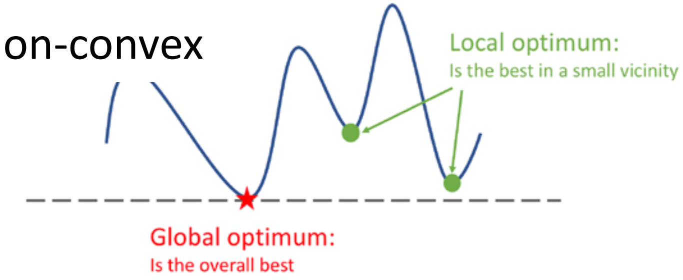

$$\min_x [f_1(x), f_2(x), \dots, f_k(x)]^T$$

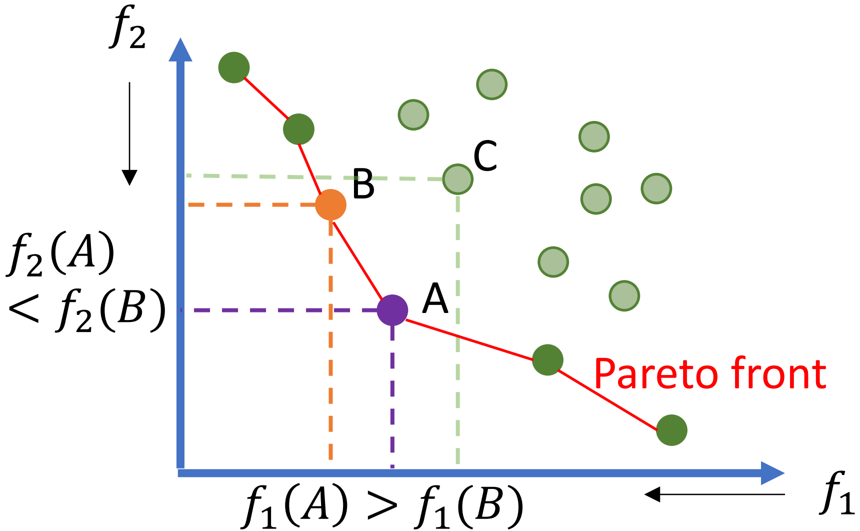
$$x \in \Omega \subset \mathbb{R}^d$$

$$[f_1, f_2, \dots, f_k]^T : \mathbb{R}^d \mapsto \mathbb{R}^k$$

$x$ → **Black-box simulation** → $[f_1(x), f_2(x), \dots, f_k(x)]^T$


Non-convex




We want to find good approximations of the Pareto front efficiently and effectively

# Reformulation methods or off the shelf evolutionary strategies are often not suitable

Linear scalarization:

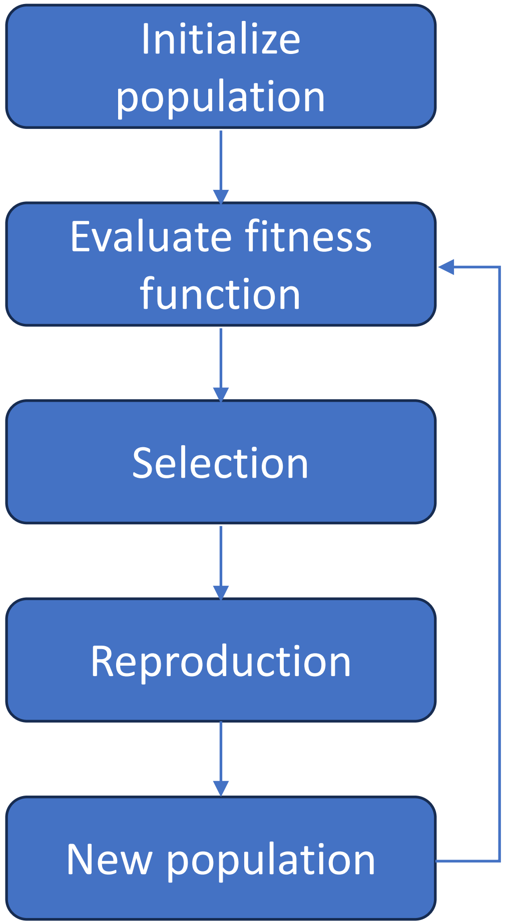
$$\min_{x} \sum_{i=1}^{k} w_i f_i(x), \quad w_i > 0$$

$\epsilon$ - constraint method

$$\min_{x} f_m(x)$$
$$\text{s.t. } f_i(x) \leq \epsilon_i, i \neq m$$

- Underlying assumptions may not be fulfilled by the problem at hand
- We get 1 solution at a time and would have to solve the problem many times to get multiple trade-off solutions

Genetic algorithm




- Requires too many expensive function evaluations off the shelf

# Surrogate models help alleviate the computational expense

A surrogate model approximates the expensive objective function:

$$f_i(x) = s_i(x) + e_i(x)$$

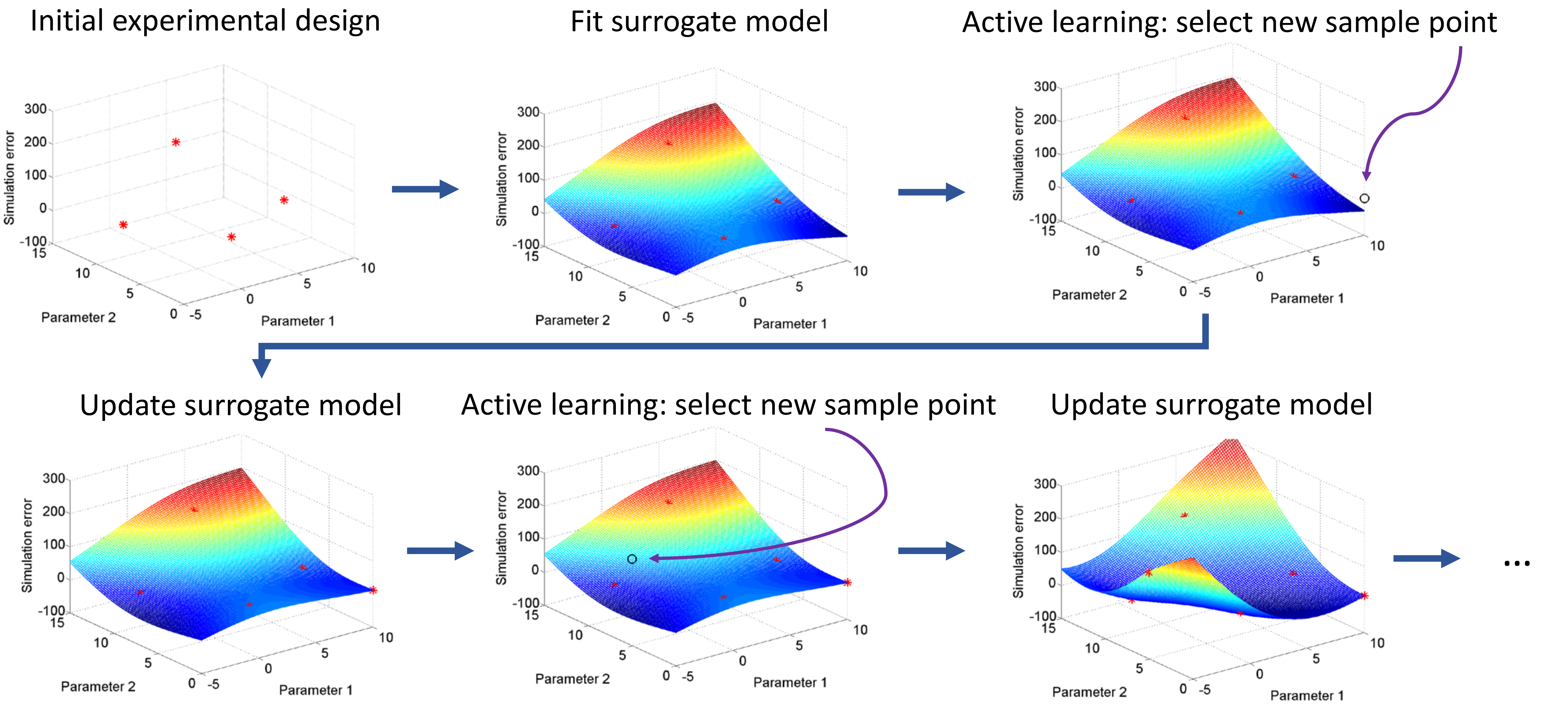
$i$-th computationally expensive objective

$i$-th computationally cheap surrogate model

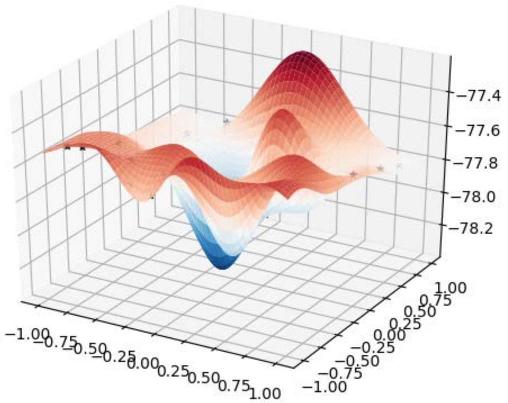
Different between the true function and the approximation

The idea is to exploit the surrogate models for guiding the optimization search and update the surrogate models each time a new input-output data pair is obtained

# Cartoon of a surrogate model based optimization algorithm (single objective)

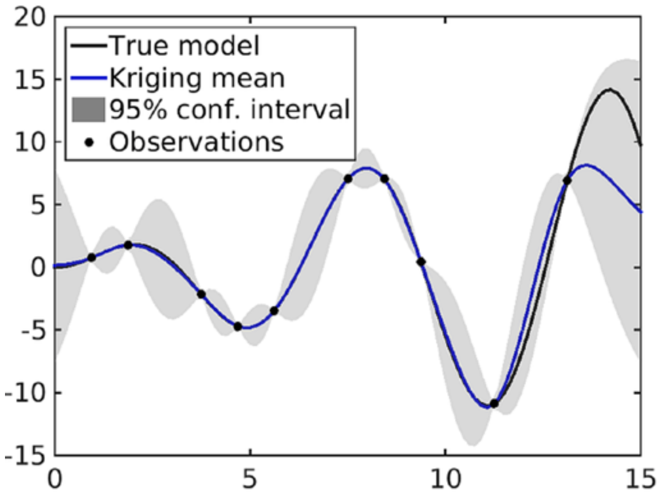


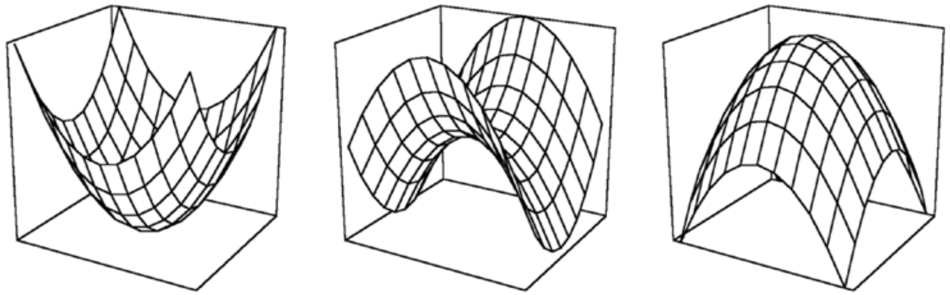
# Different types of surrogate models exist




Radial basis functions, e.g.,
$s(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|_2) + p(x)$

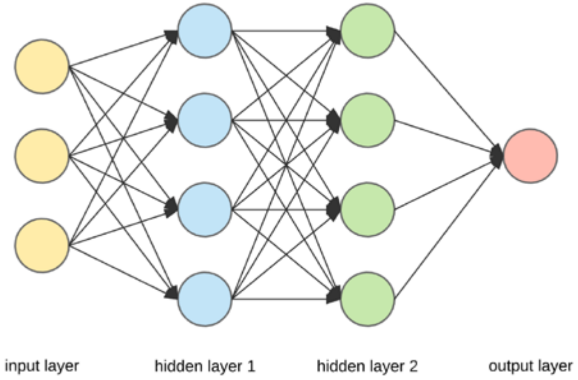
Gaussian process model,
$s(x) = \mu + Z(x), Z(x) \sim \mathcal{N}(0, \sigma^2)$



Polynomial models, e.g.,
$\mathrm{s}(x) = ax_1^2 + bx_2^2 + cx_1 + dx_2 + e$

Large data settings:
Deep Learning models, e.g.,
$\mathrm{s}(x) = \mathrm{A}[\sum w_i x_i + b]$

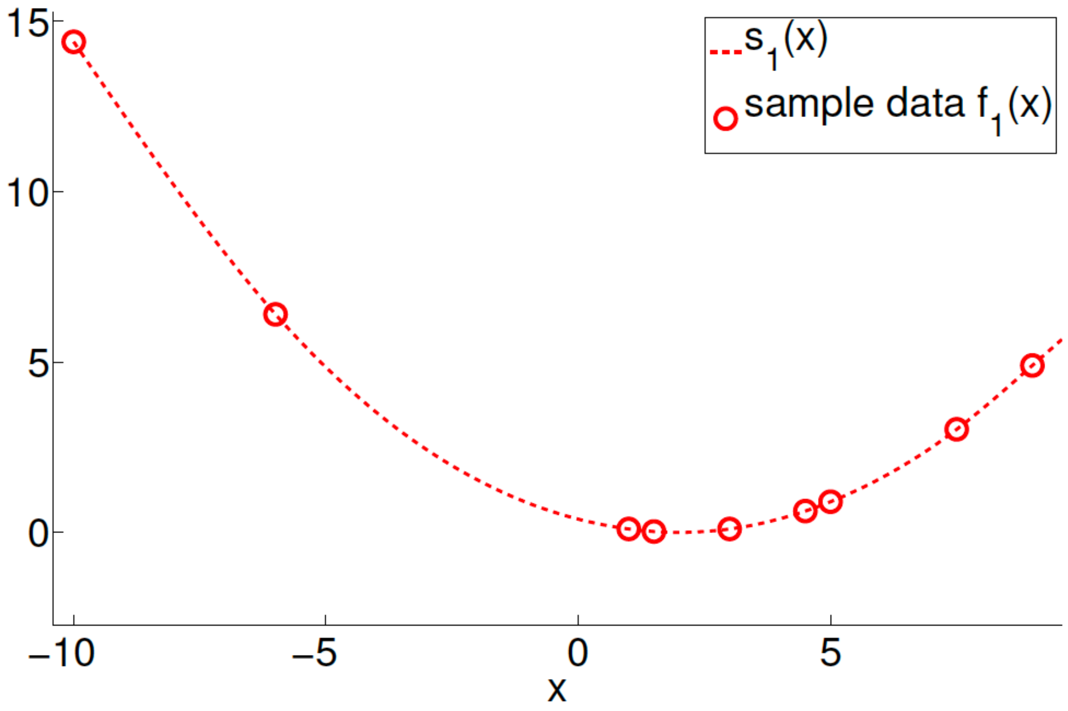


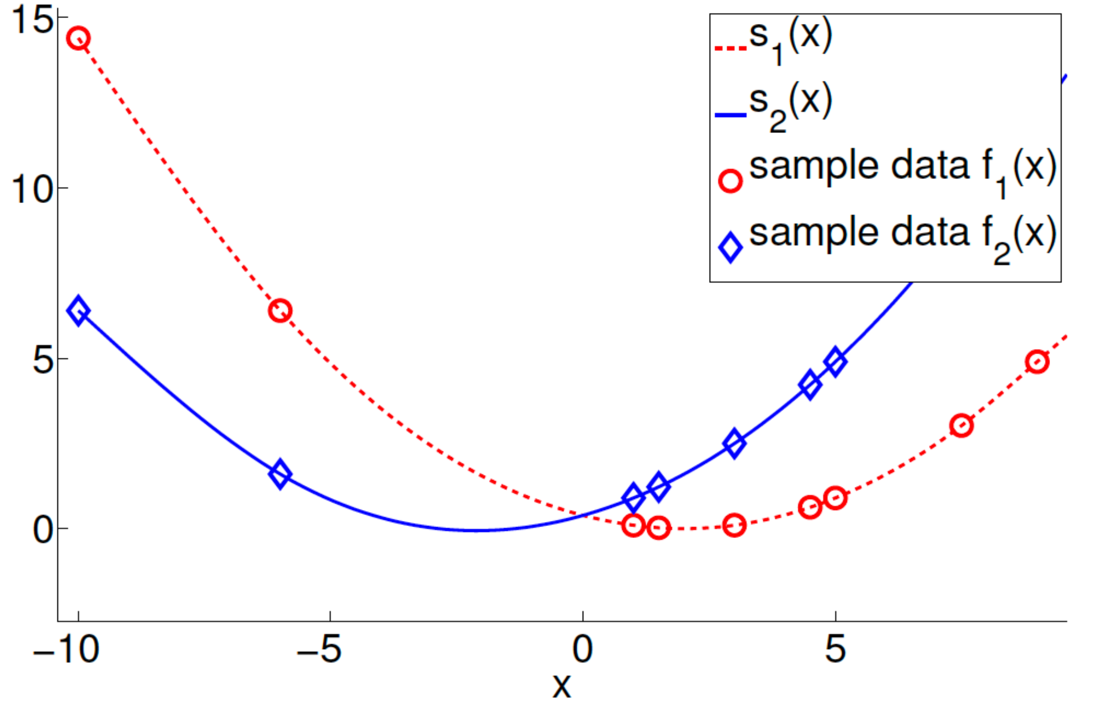

Surrogate model choice depends on the problem characteristics

# In the multi-objective setting, we fit a separate surrogate to each expensive objective function

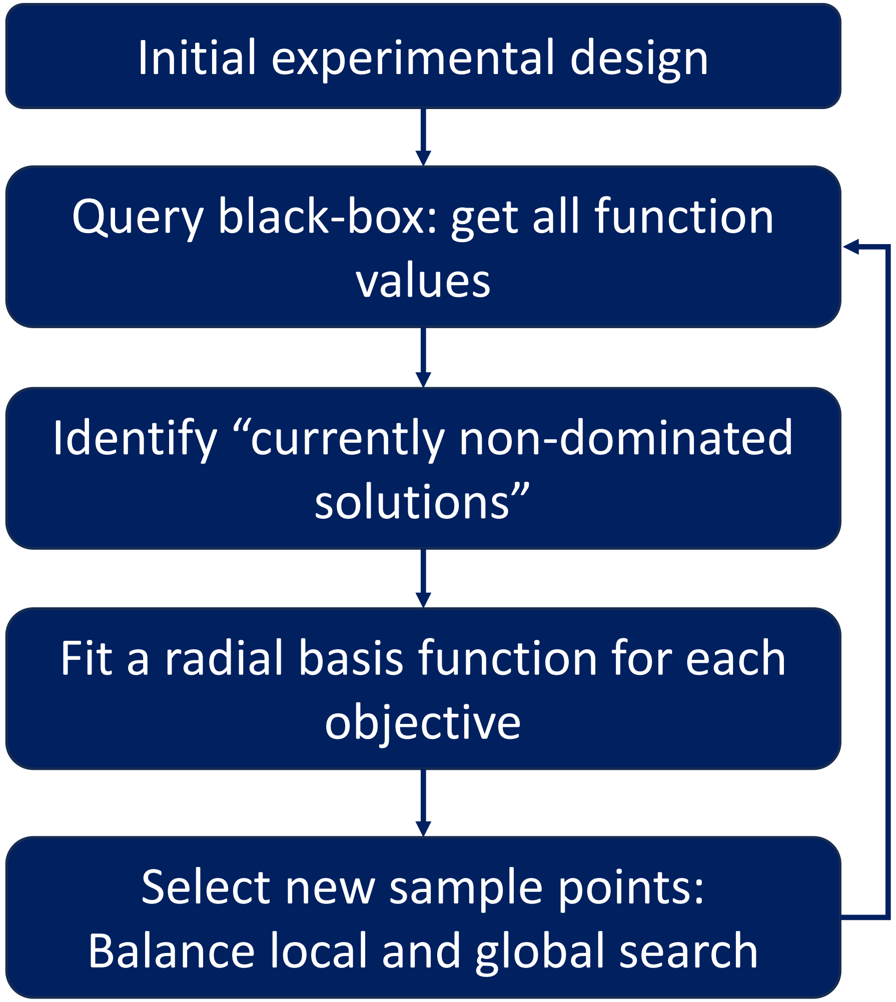
We assume that all objective functions have been evaluated at the same points in the search space:

- One call to the black box provides all objective function values



# Surrogate-guided multi-objective optimization




e.g., Latin hypercube, low discrepancy sequence

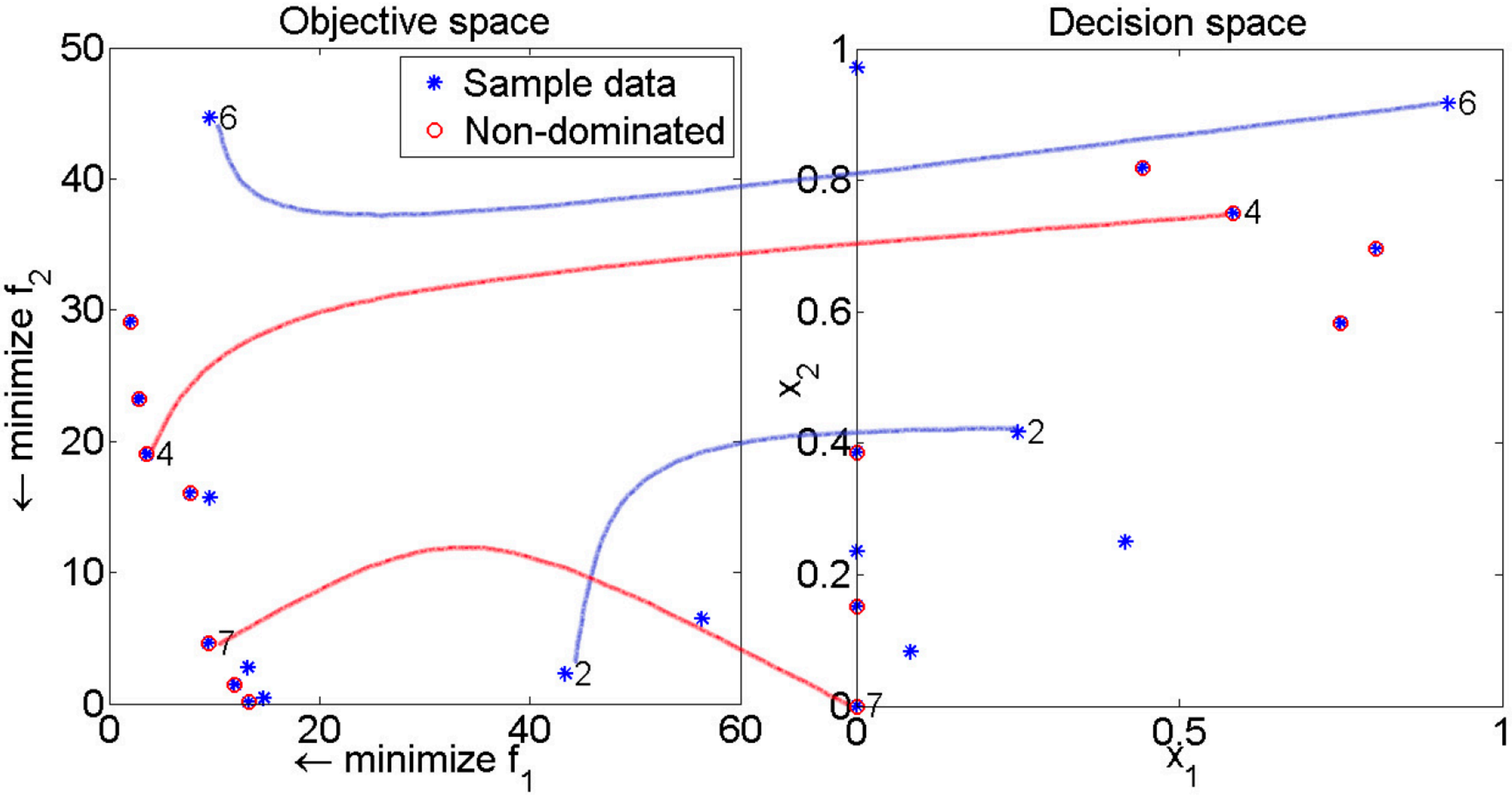
Only consider the set of evaluated points and pick non-dominated solutions

Could also fit Gaussian process, polynomial, MARS,..

Sample point selection should aim at several qualities: exploration of the PF extrema, local PF improvements, good distribution of points on the PF

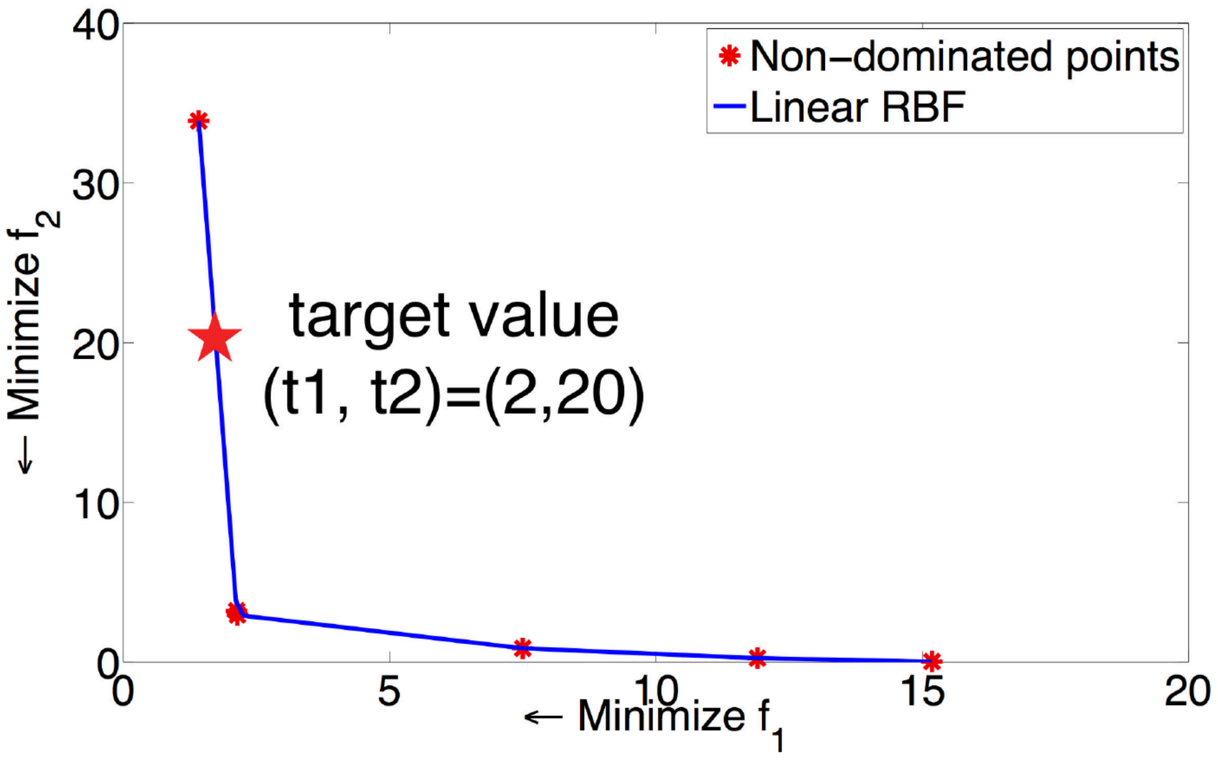
The algorithm stops when we reach a maximum budget of function evaluations

# Sampling strategies: consider both decision and objective space



# Target value based sampling

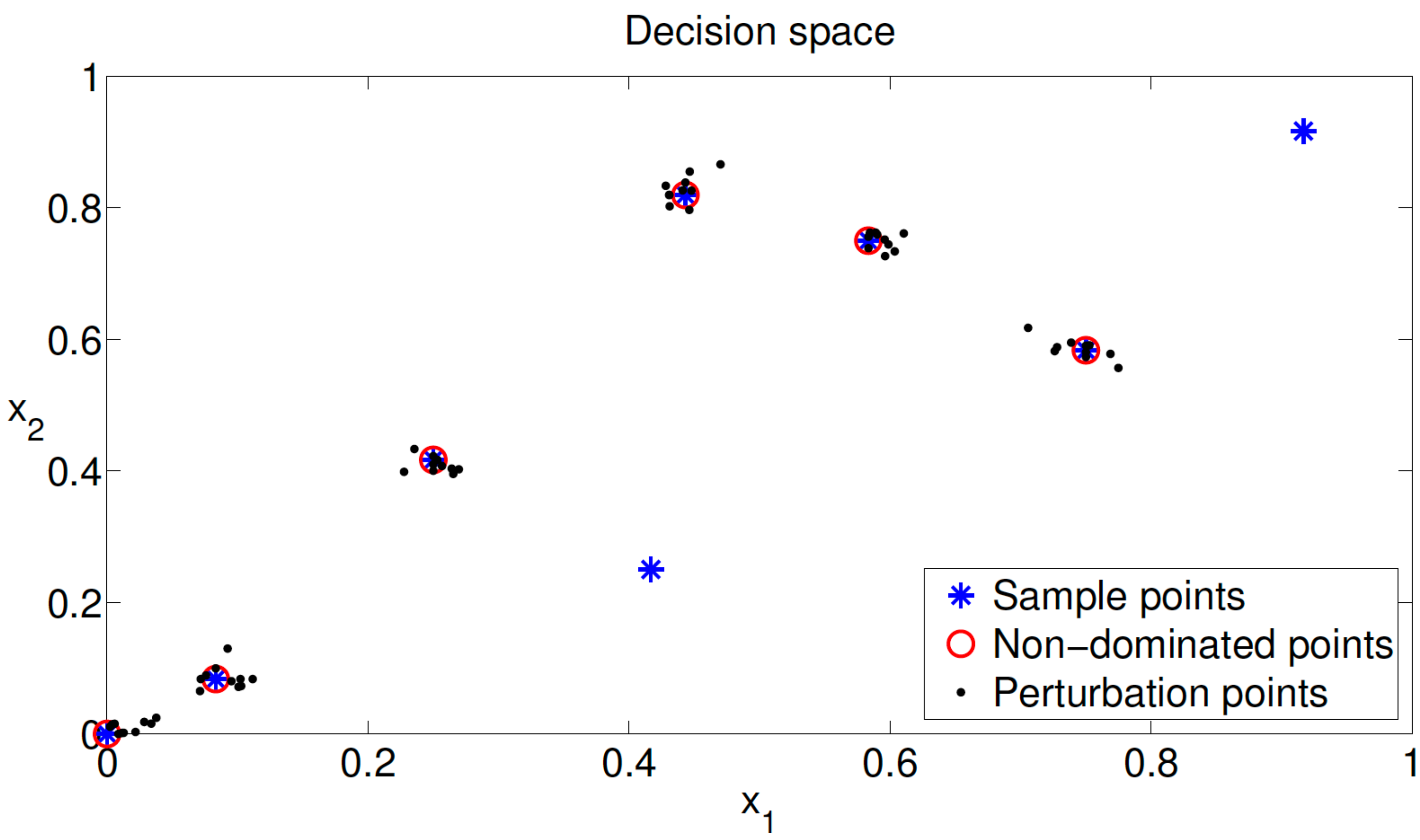
- Fit a piecewise linear function to the approximate Pareto front
- Find large gaps in the front, eg find $x$ in decision space for which $f_1(x) = 2$ and $f_2(x) = 20$ (target values)




$$\min_{x \in \Omega} [|s_1(x) - t_1|, |s_2(x) - t_2|]^T$$

- Computationally inexpensive auxiliary optimization problem using surrogate models
- Surrogate models are only approximations (there may not be a solution that minimizes both objectives)

# Perturbation of non-dominated points in decision space



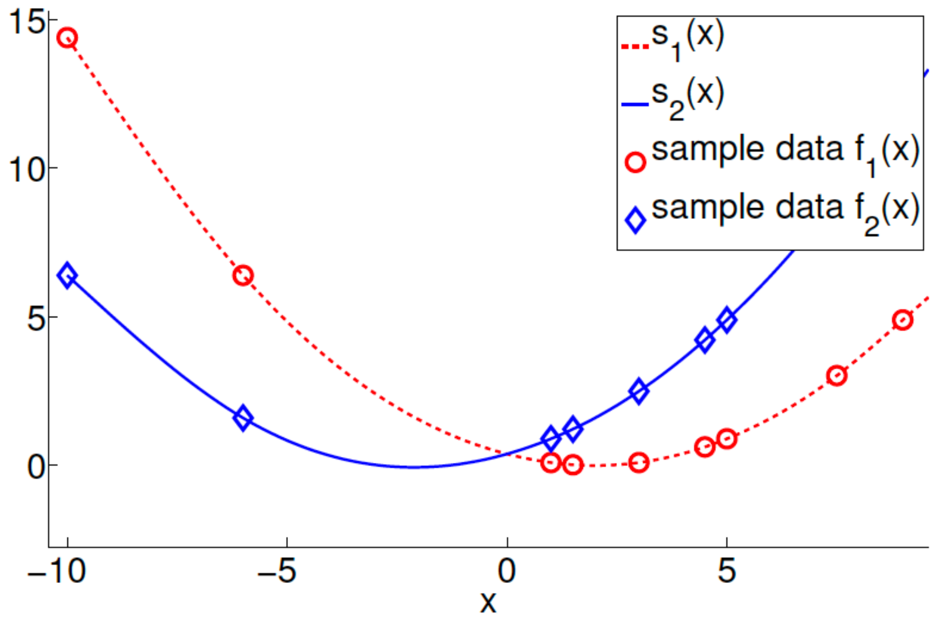

- Small perturbations of currently non-dominated points may lead to local improvements of the current Pareto-front

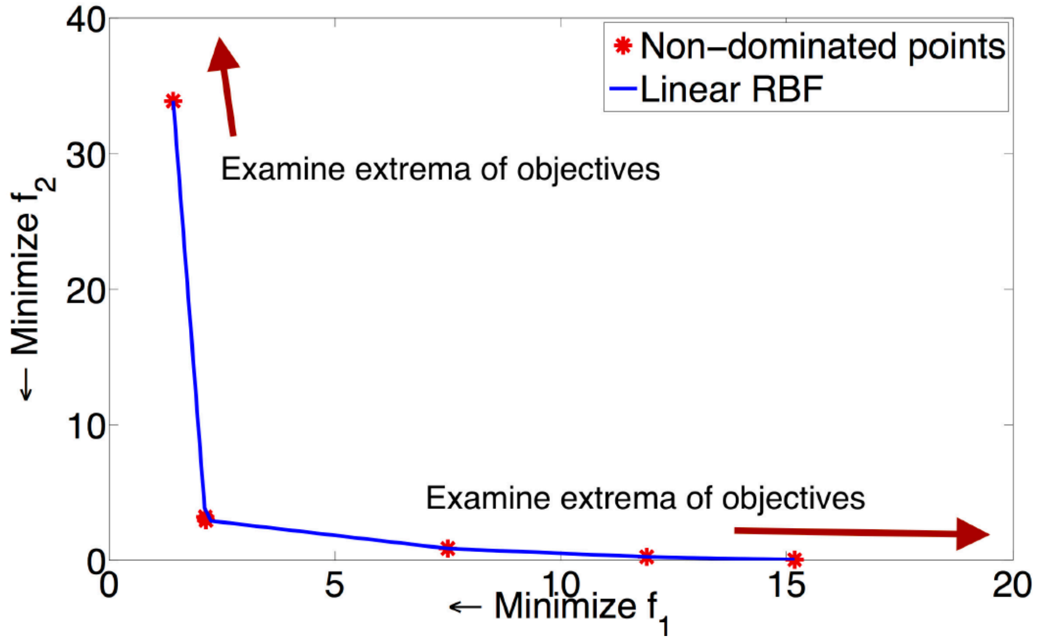
# Identify extrema of the objective functions

Minimize each objective individually using the surrogates:

$$\min_{x \in \Omega} s_i(x)$$

Using multi-start optimization for example



# Stochastic sampling and scoring

- Randomly generate points in the decision space
- Score each point:
  - Use surrogate models $s_1, \ldots s_k$ to predict each point's objective function values
  - Compute the distance of each point to the set of already evaluated points
  - Aggregate all $k+1$ values in a combined weighted score
- Select the best point as new evaluation point

# Solve surrogate multi-objective optimization problem directly

Use a genetic algorithm to solve

$$\min_{x \in \Omega} [s_1(x), s_2(x), \ldots, s_k(x)]^T$$

May result in a large number of solutions -> randomly select a subset for evaluation with the black box

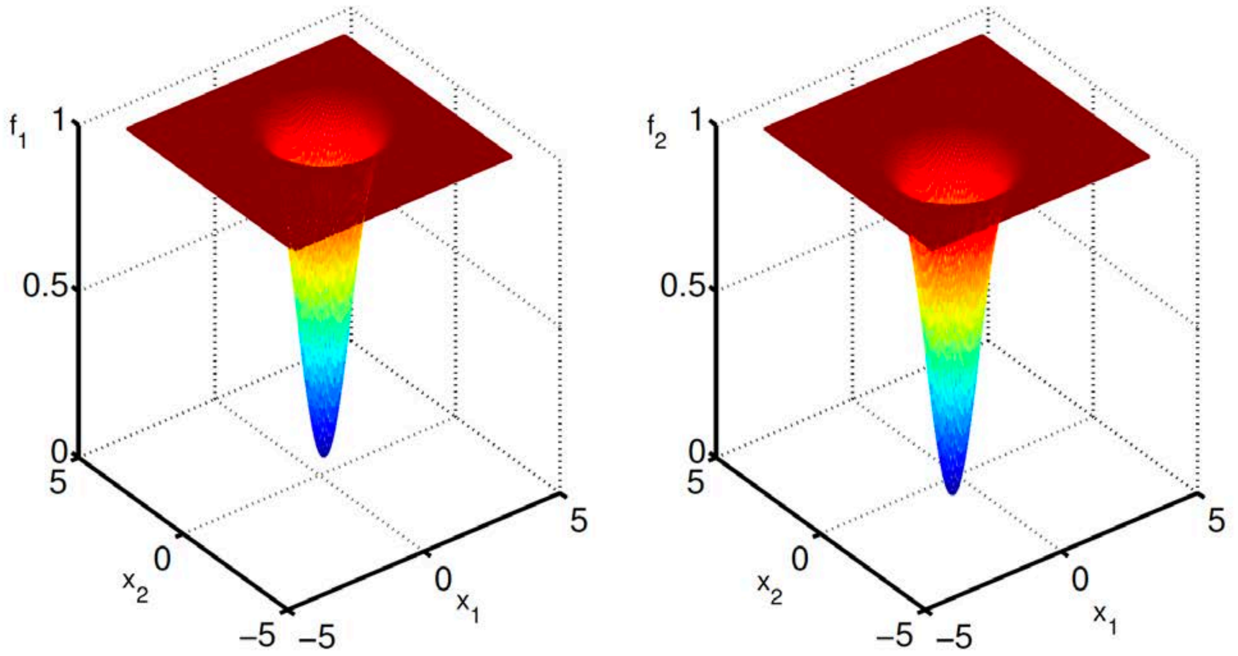
# Surrogate modeling and diverse sampling strategies work well: a little cartoon

$$\min_x [f_1(x), f_2(x)]^T$$

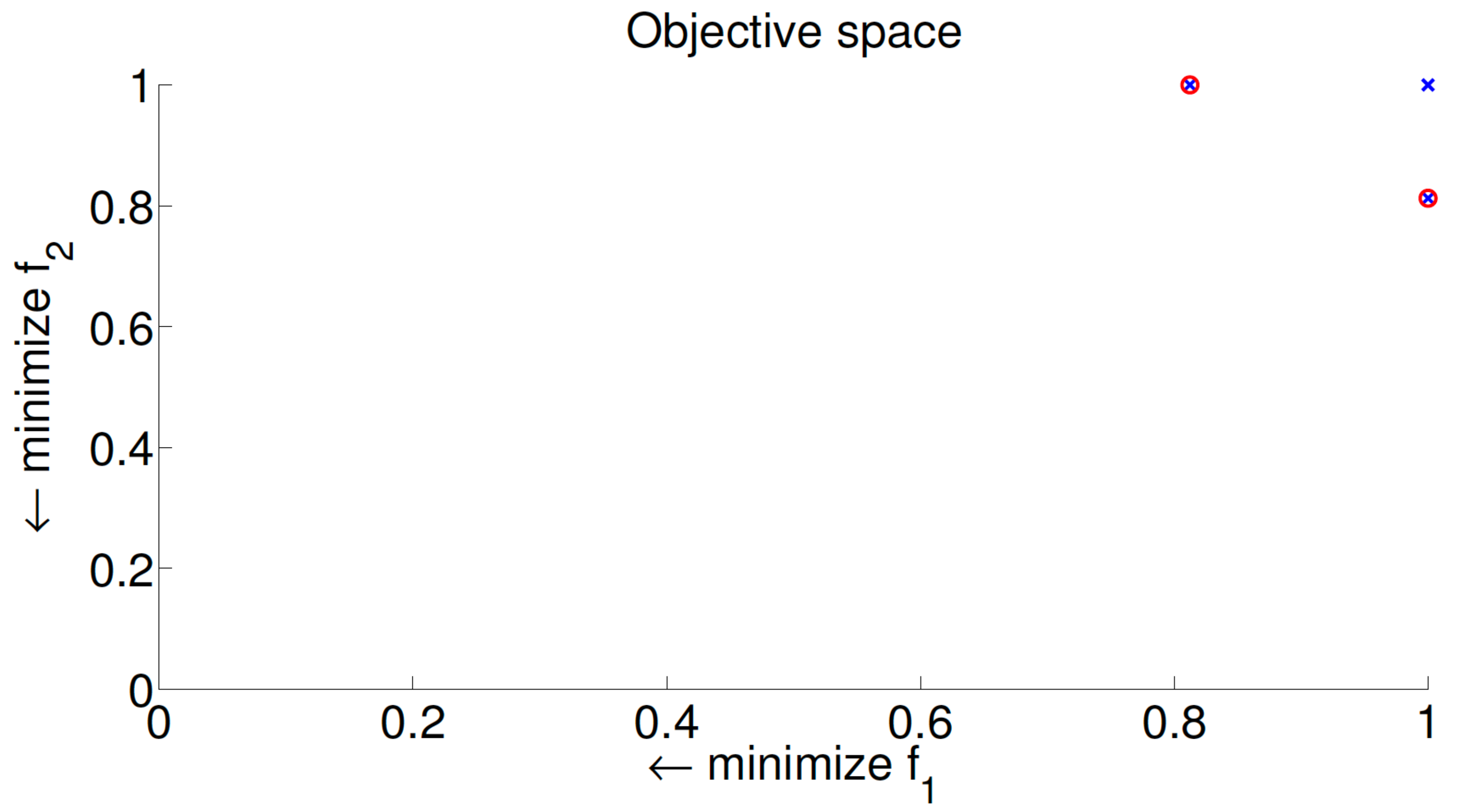
$$f_1(x_1, x_2) = 1 - exp\left\{-\sum_{j=1}^{2}\left(x_j - 1/\sqrt{2}\right)^2\right\}$$

$$f_2(x_1, x_2) = 1 - exp\left\{-\sum_{j=1}^{2}\left(x_j + 1/\sqrt{2}\right)^2\right\}$$

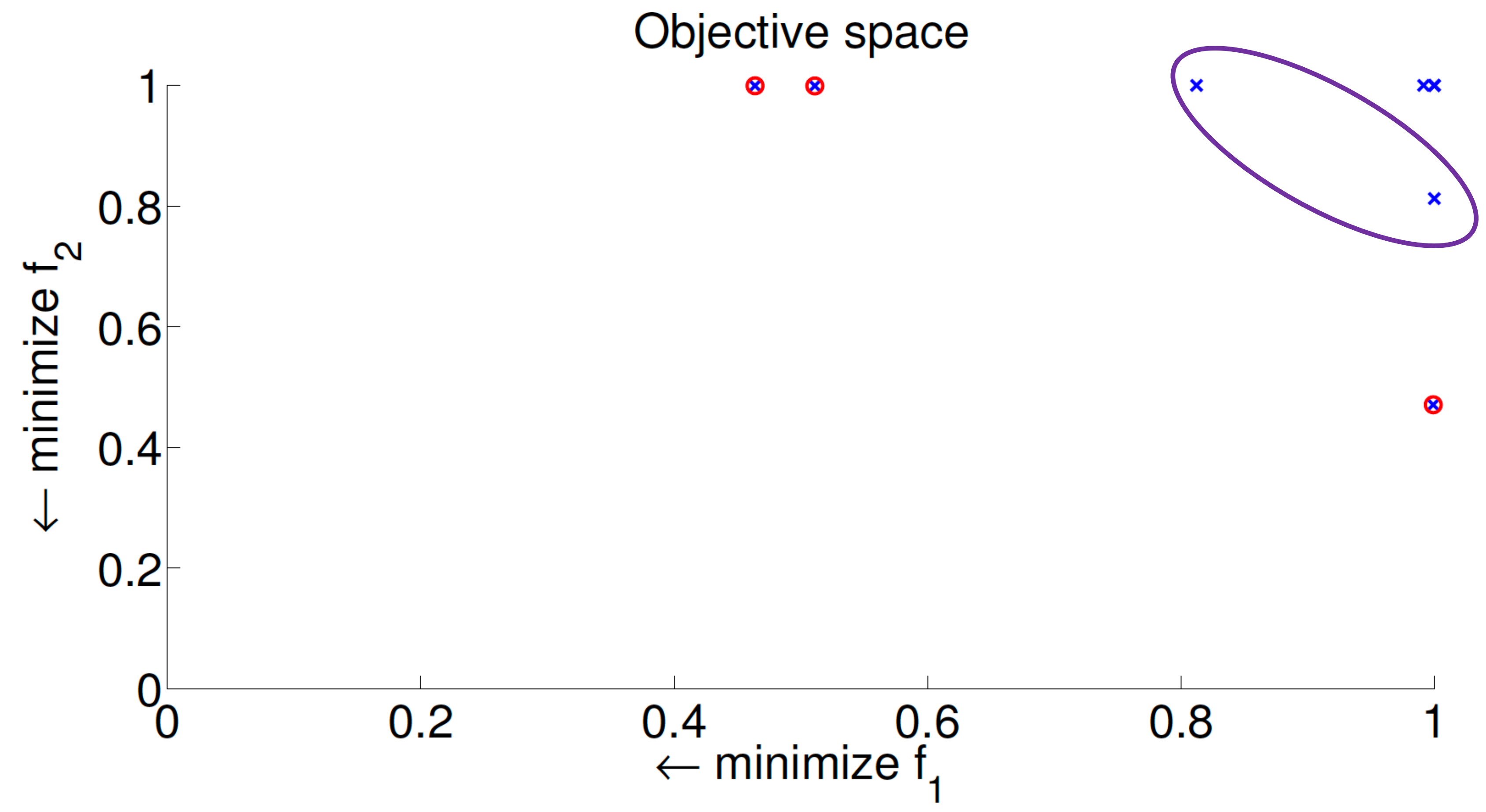
$$-4 \le x_1, x_2 \le 4$$



# Surrogate modeling and diverse sampling strategies work well: a little cartoon

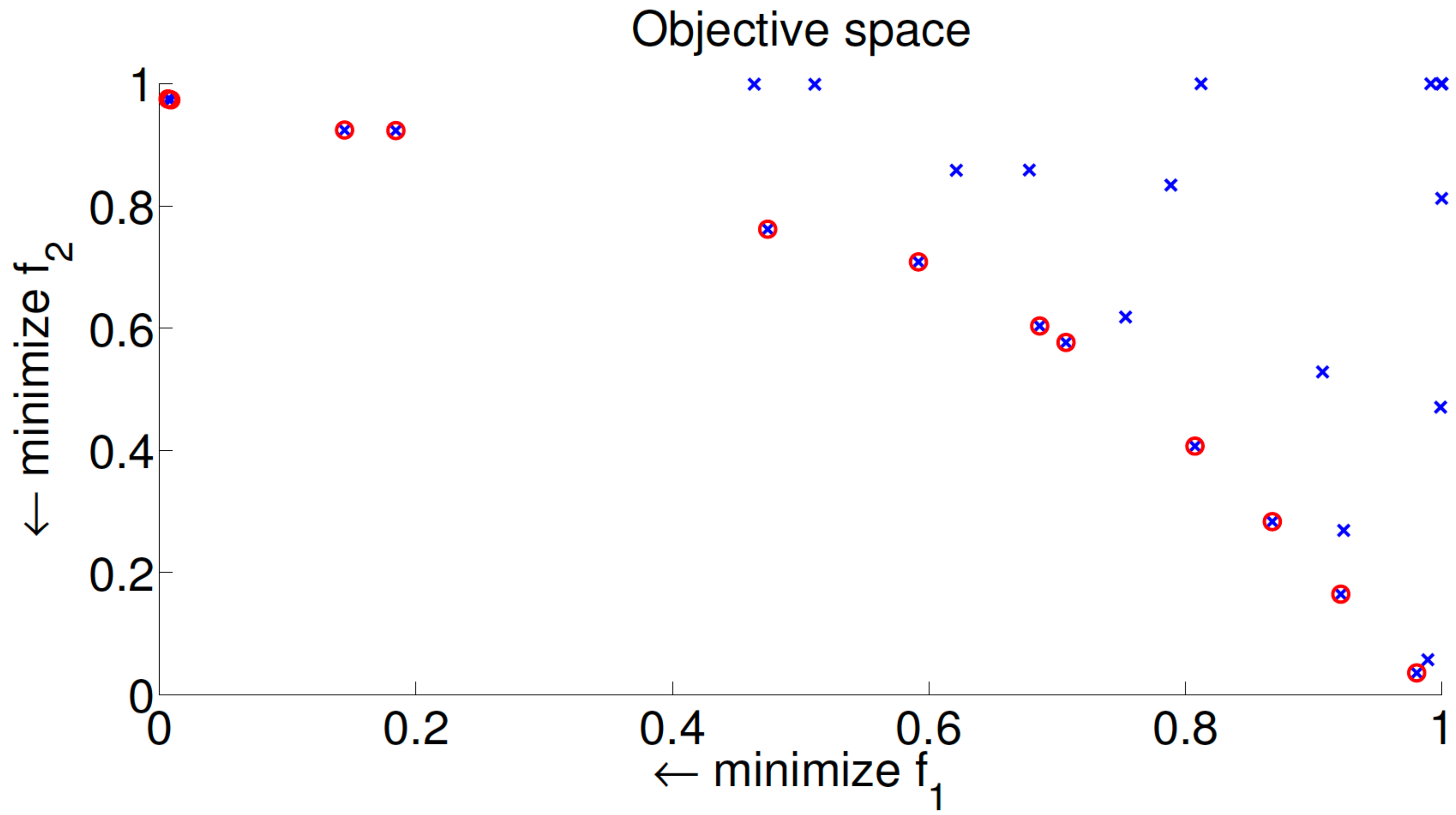


# Surrogate modeling and diverse sampling strategies work well: a little cartoon

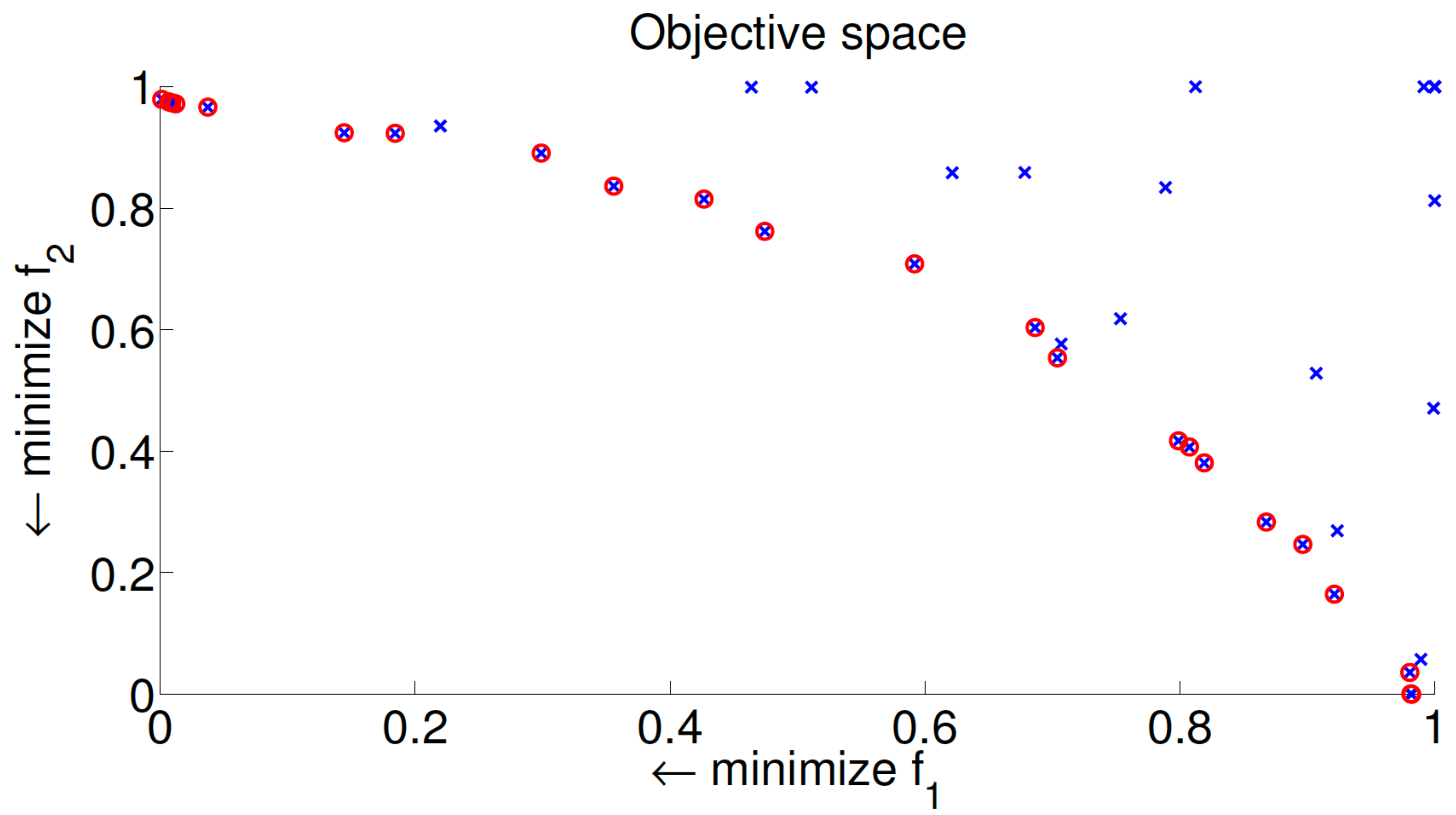



Prev non-dominated points become dominated

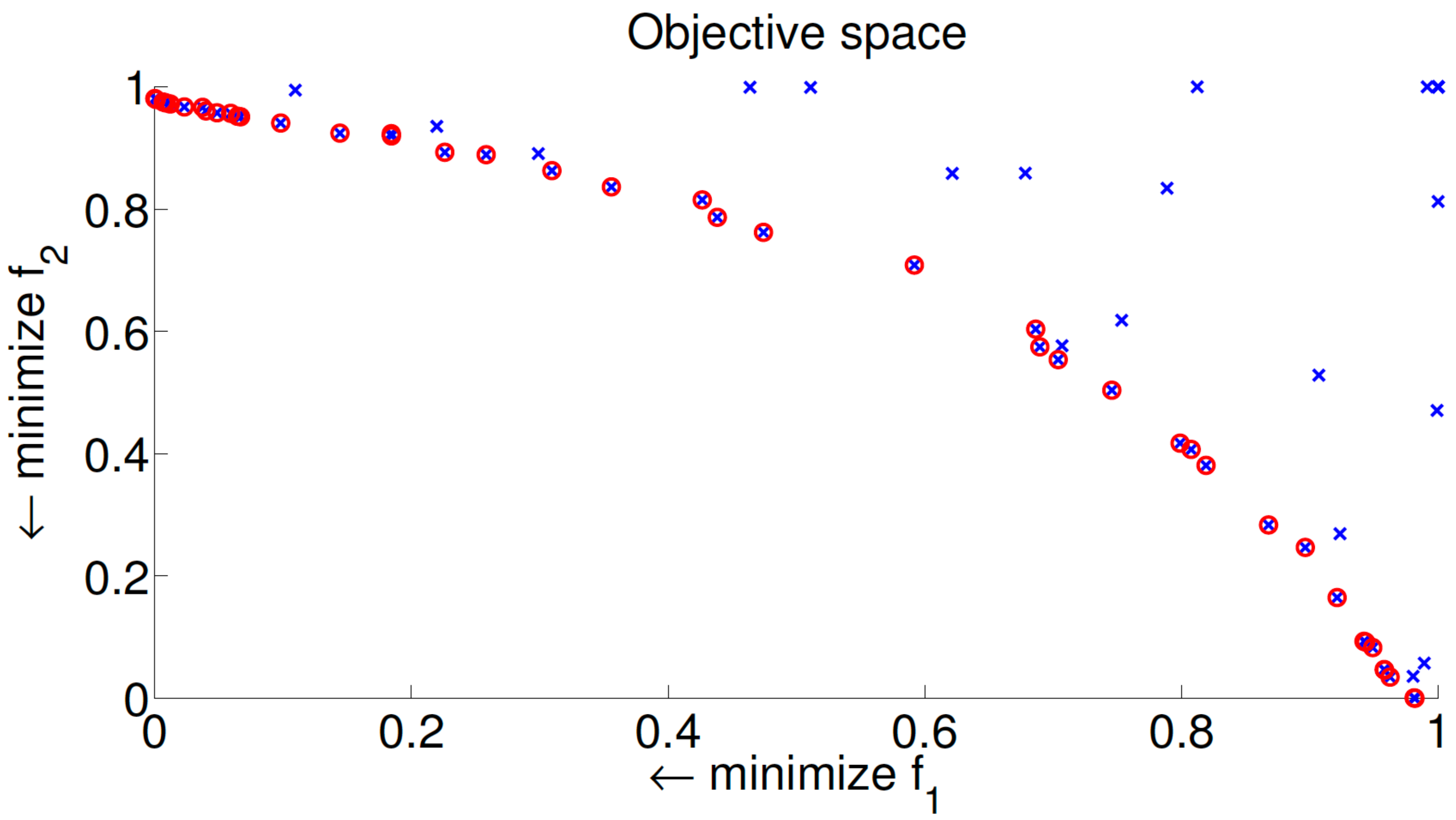
# Surrogate modeling and diverse sampling strategies work well: a little cartoon



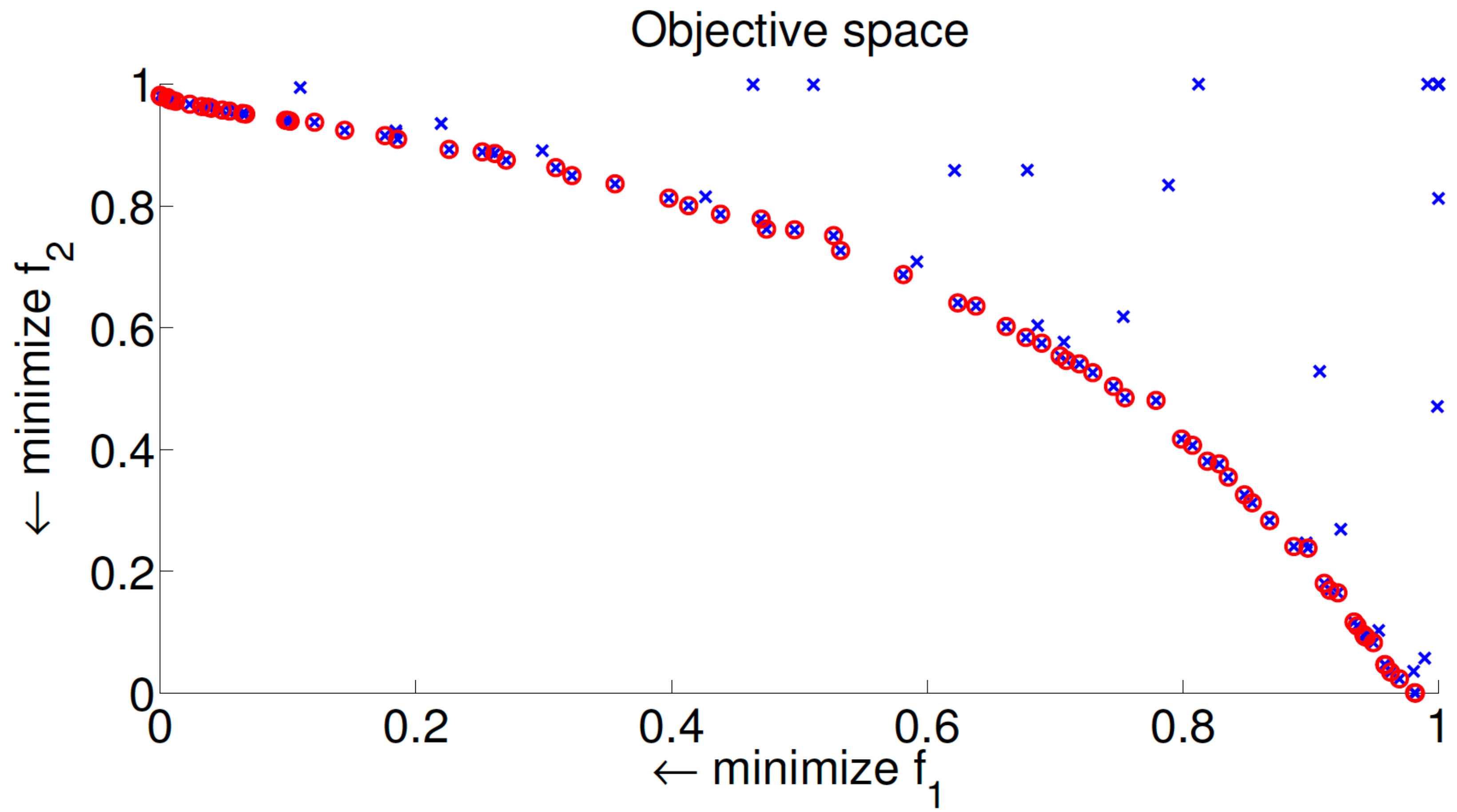
# Surrogate modeling and diverse sampling strategies work well: a little cartoon



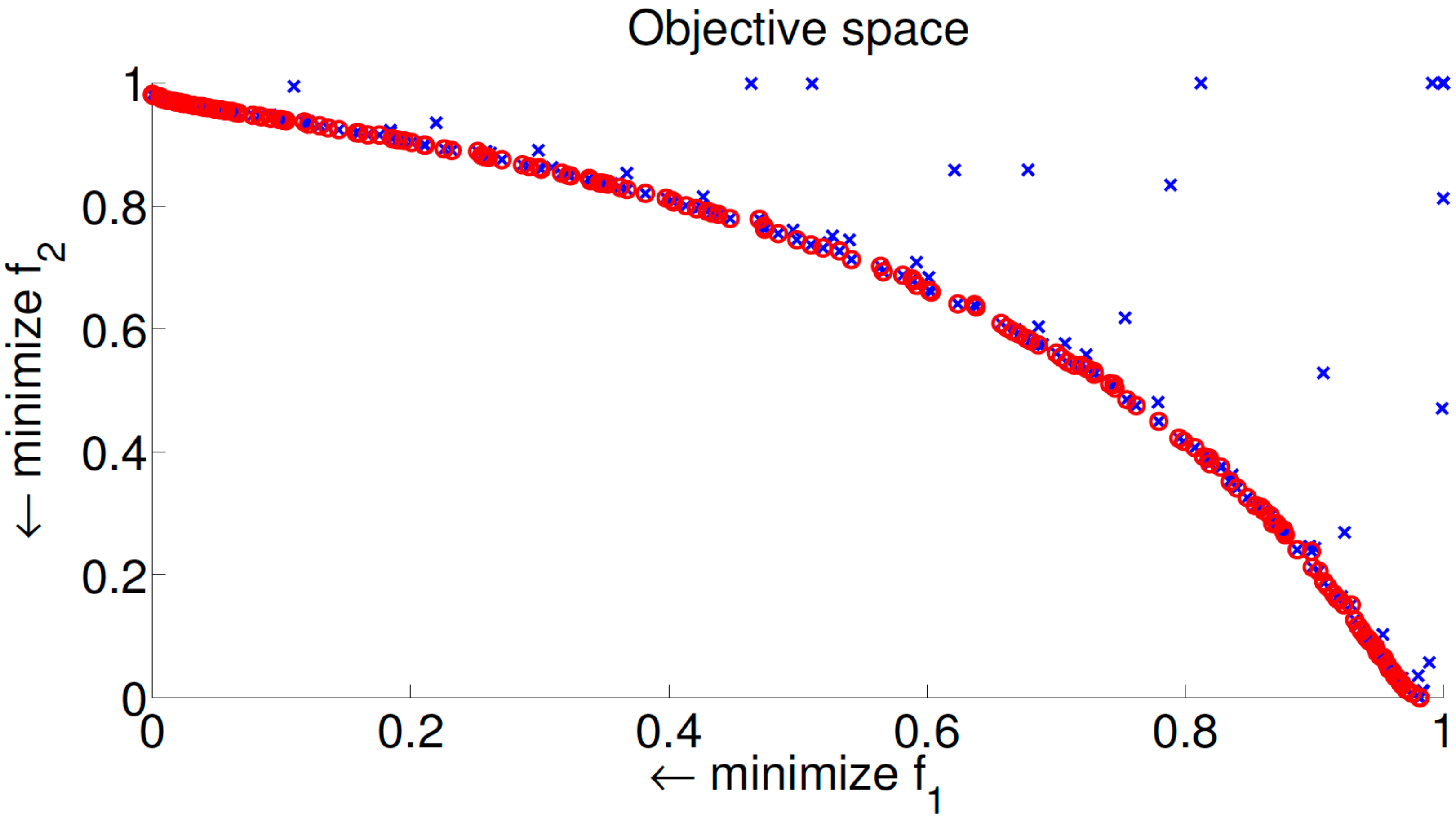
# Surrogate modeling and diverse sampling strategies work well: a little cartoon



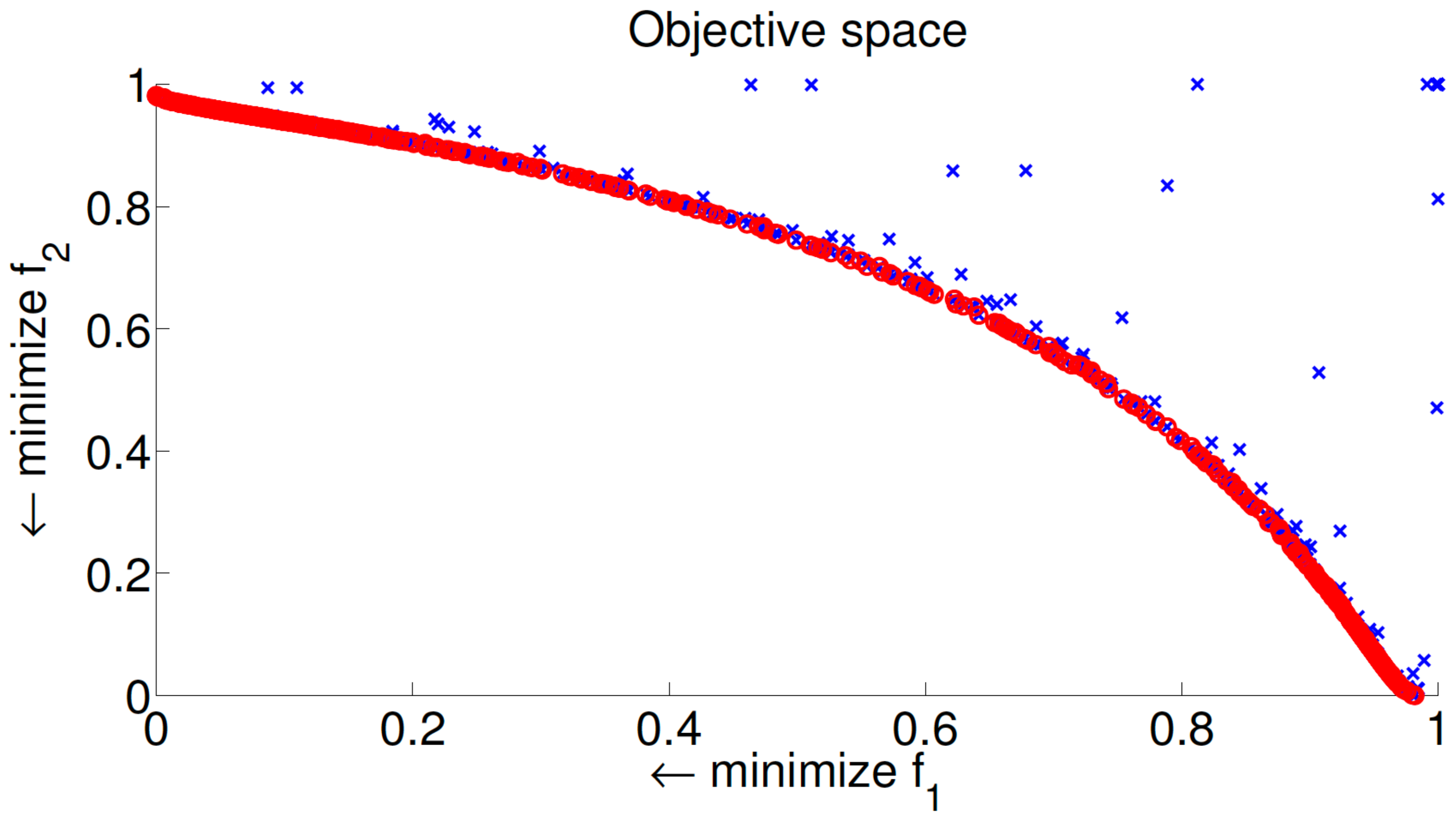
# Surrogate modeling and diverse sampling strategies work well: a little cartoon



# Surrogate modeling and diverse sampling strategies work well: a little cartoon

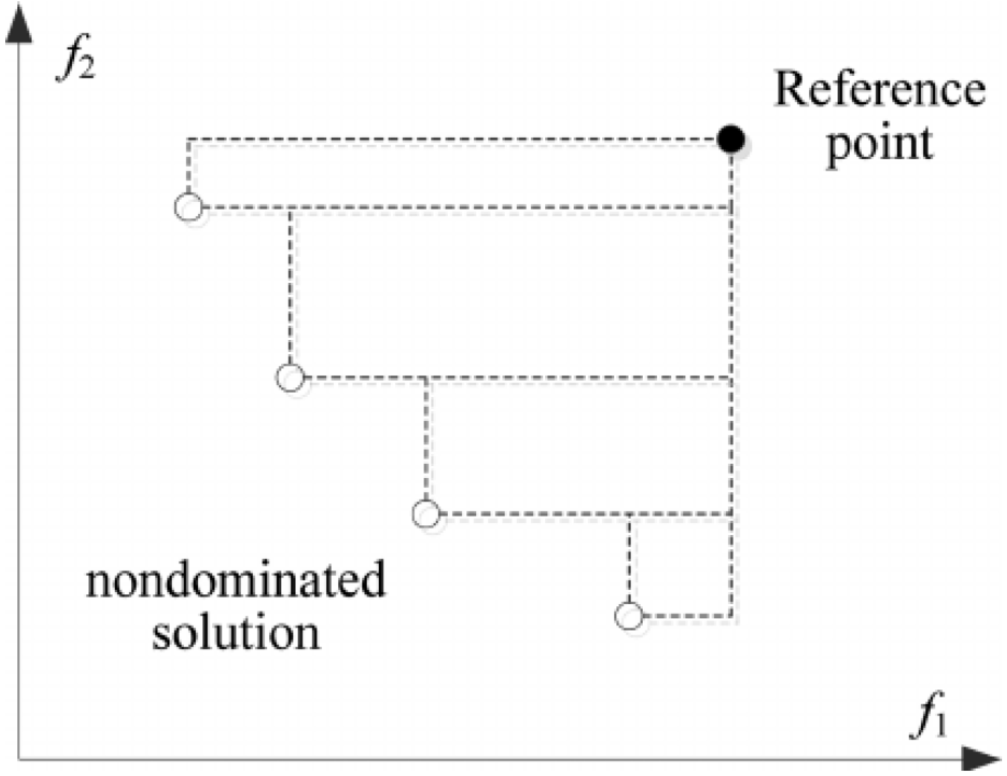


# Surrogate modeling and diverse sampling strategies work well: a little cartoon



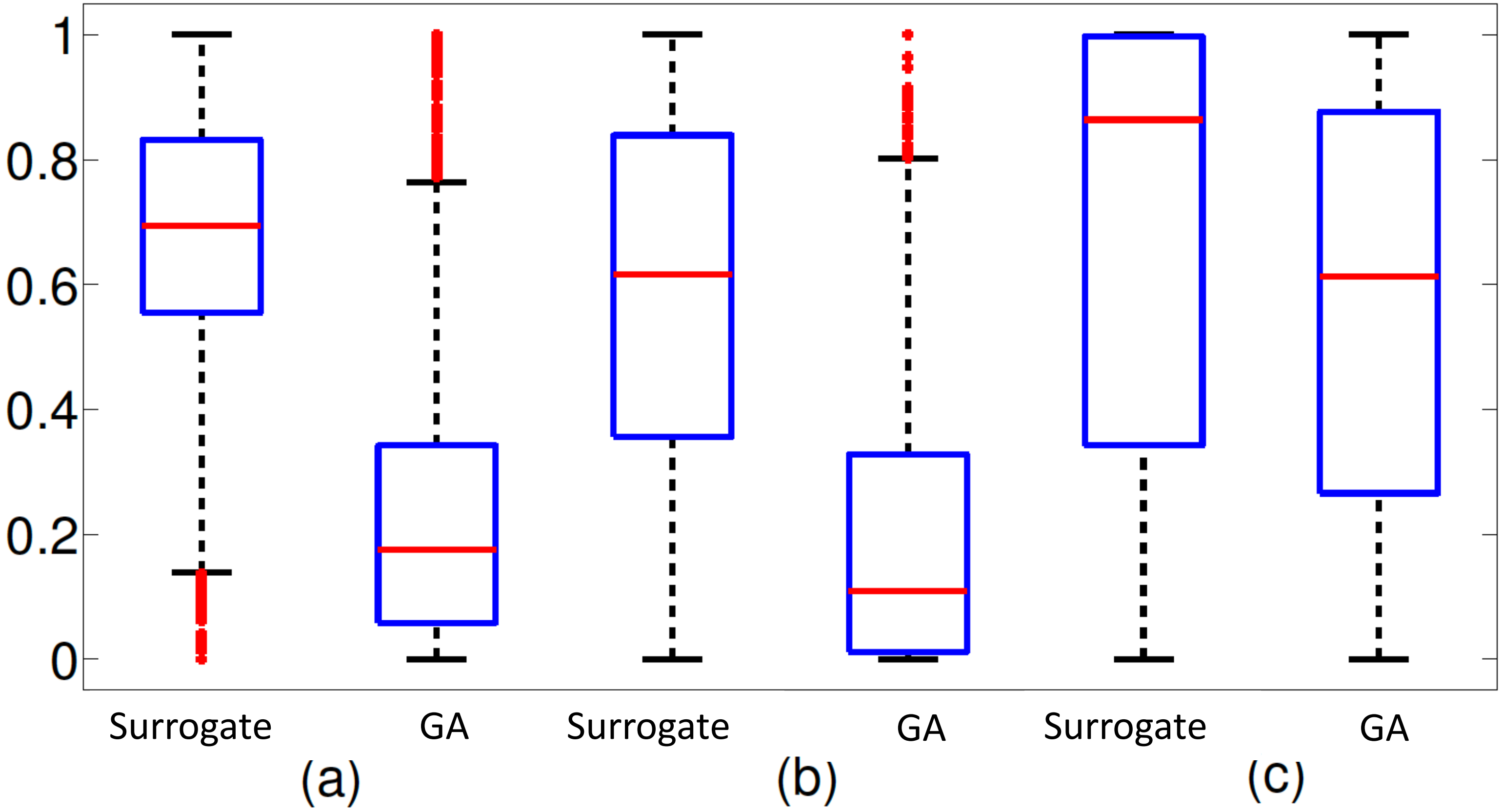
# Thorough comparison to multi-objective genetic algorithm

- Comparison metrics:
  - # non-dominated solutions
  - Set coverage (proportions of MOGA solution dominated by SO and vice versa)
  - hypervolume




- 58 benchmark problems includes
  - 1 airfoil design app
  - 1 structural engineering app
  - 1-35 decision variables
  - 2-10 objective functions
  - Pareto fronts: Convex connected, concave connected, disconnected, unknown

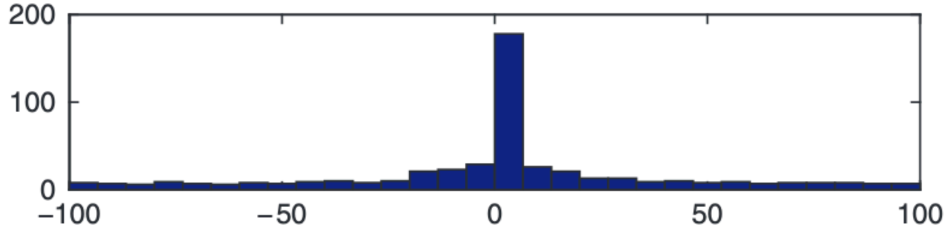
# Results overview over all problems: larger numbers are better



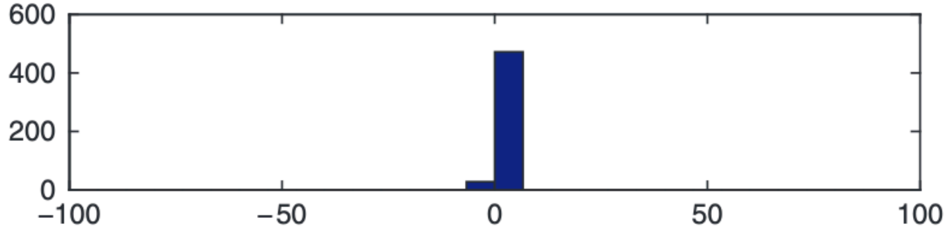

(a) Number non-dominated solutions
(b) Set coverage metric
(c) hypervolume

# Surrogate optimizer and GA sample at different points

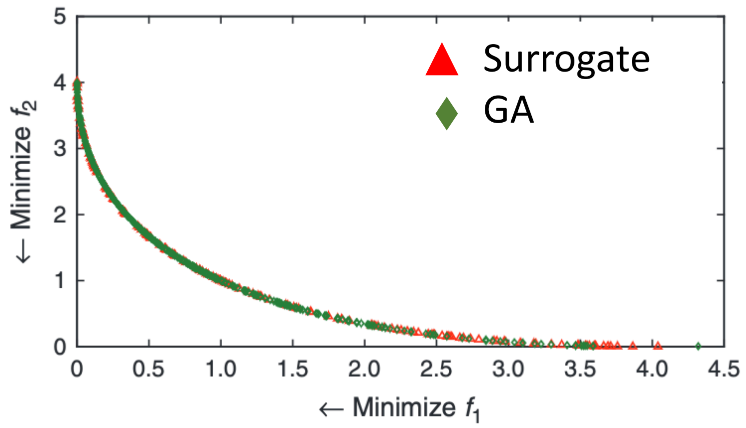
## Convex connected



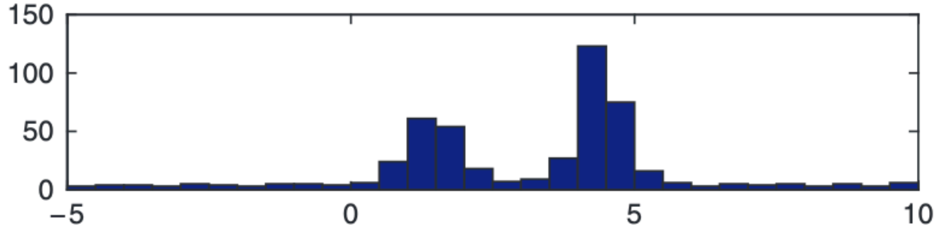
$x$ values sampled by surrogate optimizer



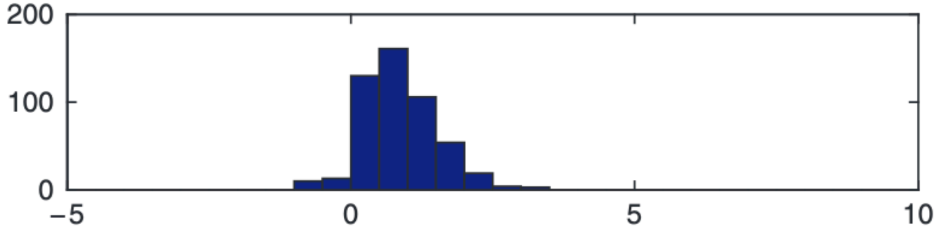
$x$ values sampled by GA



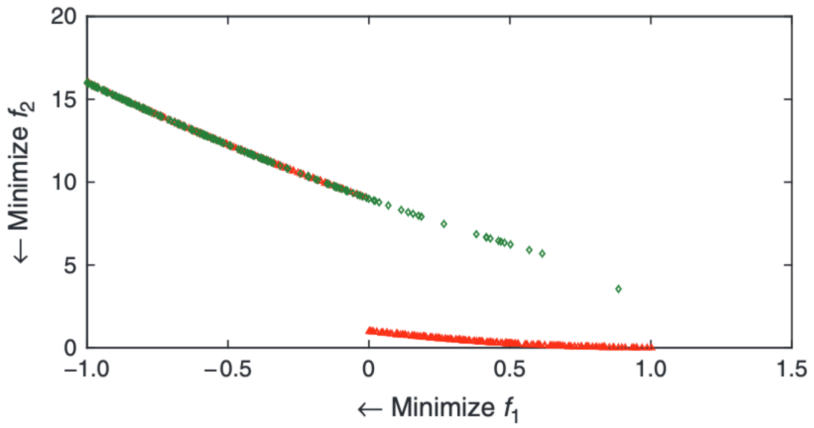

## Disconnected



$x$ values sampled by surrogate optimizer

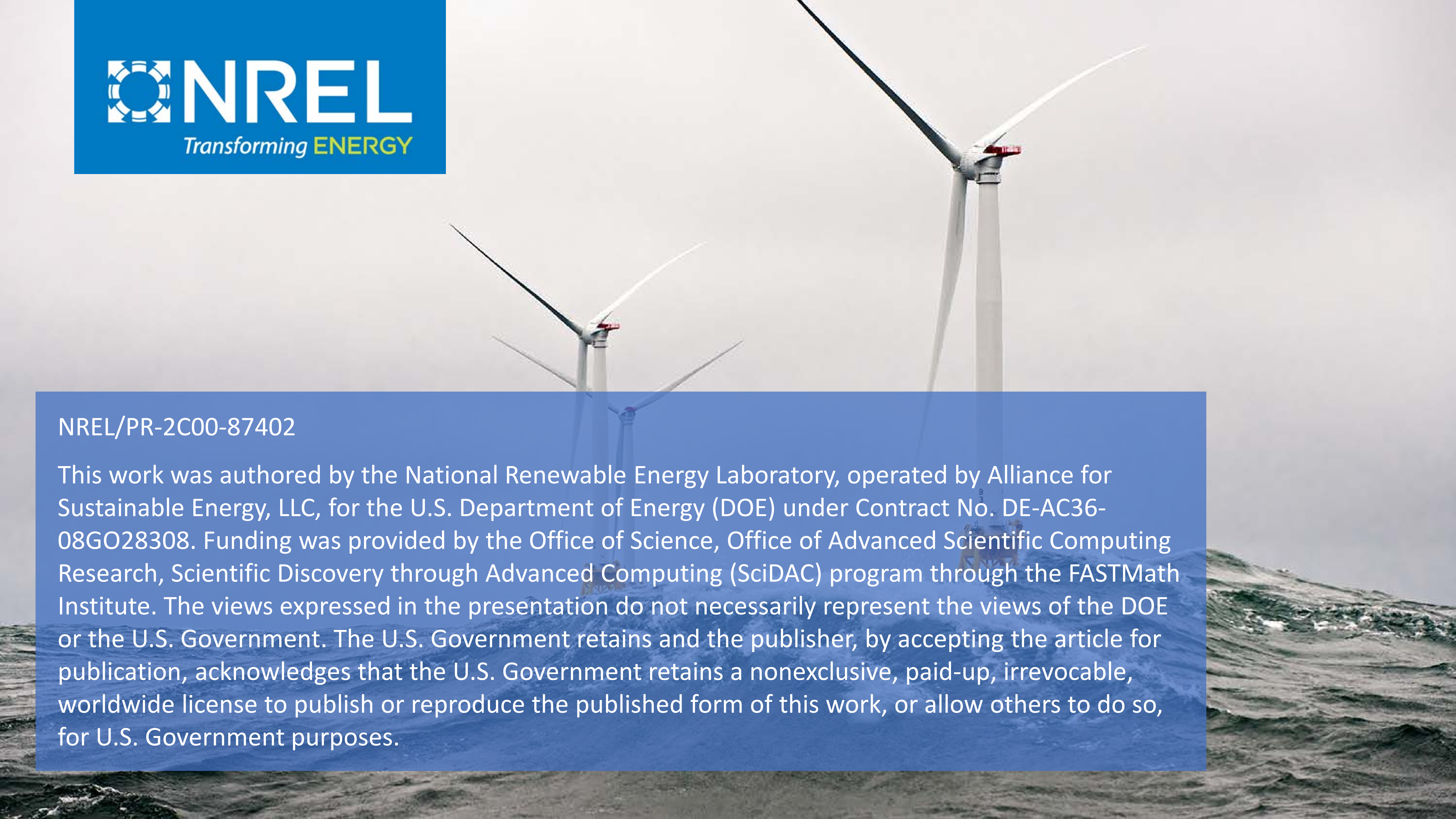


$x$ values sampled by GA



# Some things to think about

- **Convergence**: can we get anything better than convergence in probability?
  - Practical algorithms vs convergence proofs?
  - We can develop loads of sample strategies, can we develop guarantees or rules as to when to use which?
- **Noisy functions**: how do we define Pareto optimality when function evaluations are noisy? Can we adapt current algorithms to noisy problems?
  - Limited compute budget -> can't evaluate each point 30 times to get a good estimate
- Incorporating **constraints**?
- Function evaluations come from **different black-boxes** (some faster, some slower) -> "early stopping" possible?
- Incorporate **multi-fidelity information**?
- **Large dimensions** (possibly independent of multi-objective…)
- Performance metrics vs comparing algorithms (if I optimize for hypervolume, I probably will be better than anyone who didn't optimize for hypervolume – "objective"(?) comparison metrics?



NREL/PR-2C00-87402

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding was provided by the Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program through the FASTMath Institute. 