Adaptive Sampling of Pareto Frontiers with Binary Constraints using Regression and Classification

Fraunhofer ITWM

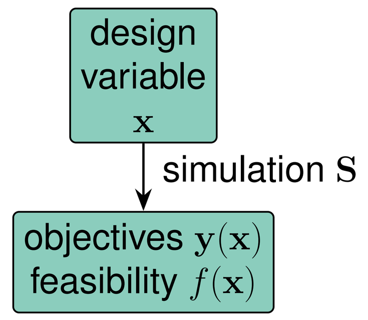
RAOUL HEESE, MICHAEL BORTZ
raoul.heese@itwm.fraunhofer.de
arxiv.org/abs/2008.12005

1. Problem description

TASK: SOLVE AN OPTIMIZATION PROBLEM!

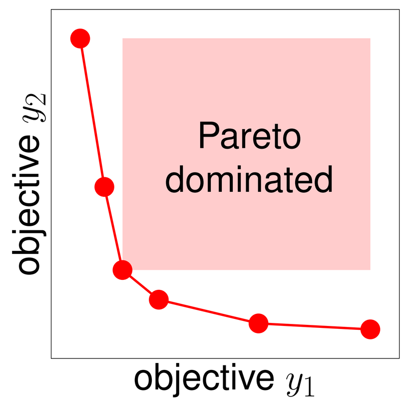
Presume a black-box computer simulation $\mathrm{S}(\mathbf{x})$ with

- real design variables $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ of arbitrary dimensions,
- multiple real black-box objectives $\mathbf{y} \in \boldsymbol{\mathcal{Y}}$ and
- a binary black-box feasibility f.



BLACK-BOX OPTIMIZATION PROBLEM:

minimize$_{\mathbf{x}}$ $\mathbf{y} \equiv \mathbf{y}(\mathbf{x}) \equiv (y_1(\mathbf{x}), \dots, y_n(\mathbf{x}))$
subject to $f \equiv f(\mathbf{x}) = \text{feasible}$
where $\mathbf{x} \in \boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^d$ (design variables)
$\mathbf{y} \in \boldsymbol{\mathcal{Y}} \subseteq \mathbb{R}^n$ (objectives)
$f \in \mathcal{F} \equiv \{\text{feasible}, \text{infeasible}\}$



FORMAL SOLUTION:

Set of Pareto optimal objectives

$$\mathbf{P}(\boldsymbol{\mathcal{X}}, \mathbf{y}(\mathbf{x}), f(\mathbf{x})) \equiv \{\mathbf{y} \in \boldsymbol{\mathcal{Y}} \mid \exists \mathbf{x} \in \boldsymbol{\mathcal{X}} : \mathbf{y} = \mathbf{y}(\mathbf{x}) \wedge f(\mathbf{x}) = \text{feasible} \wedge \mathbf{y}' \npreceq \mathbf{y} \, \forall \mathbf{x}' \in \boldsymbol{\mathcal{X}} \setminus \{\mathbf{x}\} : \mathbf{y}' = \mathbf{y}(\mathbf{x}') \wedge f(\mathbf{x}') = \text{feasible}\}$$

with $\mathbf{y} \preceq \mathbf{y}' \Leftrightarrow y_i \leq y_i' \forall i = 1, \dots, n \wedge \mathbf{y} \neq \mathbf{y}'$ ($\mathbf{y} \in \boldsymbol{\mathcal{Y}}$ dominates $\mathbf{y}' \in \boldsymbol{\mathcal{Y}}$).

GOAL:

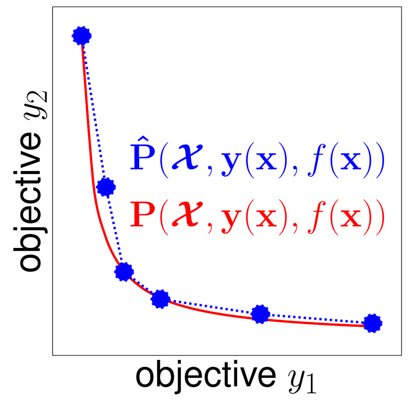
Find an approximate solution

$$\hat{\mathbf{P}}(\boldsymbol{\mathcal{X}}, \mathbf{y}(\mathbf{x}), f(\mathbf{x})) \approx \mathbf{P}(\boldsymbol{\mathcal{X}}, \mathbf{y}(\mathbf{x}), f(\mathbf{x}))$$

as accurate as possible with as few evaluations

$$\mathbf{S}(\mathbf{x}) \equiv (\mathbf{y}(\mathbf{x}), f(\mathbf{x}))$$

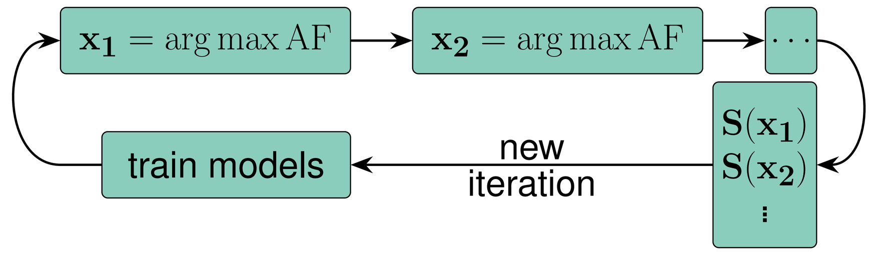
as possible.



2. Proposed method

BASED ON BAYESIAN OPTIMIZATION (BO):

- BO is a sequential design strategy for global optimization.
- In each iteration, multiple design points $\mathbf{x}_1, \mathbf{x}_2, \dots$ are proposed.
- The points are chosen based on the estimated global maximum of a so-called acquisition function AF, which is based on surrogate models for S.
- S can be evaluated simultaneously for all proposed design points.
- The resulting data is used to refine the surrogate models.



ACQUISITION FUNCTION:

$$\mathrm{AF}(\mathbf{x}) \equiv \frac{w_{\mathrm{opt}}}{|\mathbf{w}|_1} U_{\mathrm{opt}}(\mathbf{x}, \hat{\mathbf{y}}, \hat{f}) + \frac{w_{\mathrm{con}}}{|\mathbf{w}|_1} U_{\mathrm{con}}(\mathbf{x}, \hat{\mathbf{y}}, \hat{f}) + \frac{w_{\mathrm{exp}}}{|\mathbf{w}|_1} U_{\mathrm{exp}}(\mathbf{x}, \hat{\mathbf{y}})$$

- 3 components with user-defined weights $\mathbf{w} \equiv (w_{\mathrm{opt}}, w_{\mathrm{con}}, w_{\mathrm{exp}})$:
  - U_{opt}: Expected improvement of the Pareto frontier.
  - U_{con}: Expected improvement of the classification model.
  - U_{exp}: Expected exploration benefit.
- The weighted sum evaluates the *expected utility* of a design point as a customizable trade-off between exploitation and exploration.
- Based on two surrogate models for S:
  - Regression model $\hat{\mathbf{y}} \approx \mathbf{y}$
  - Classification model $\hat{f} \approx f$

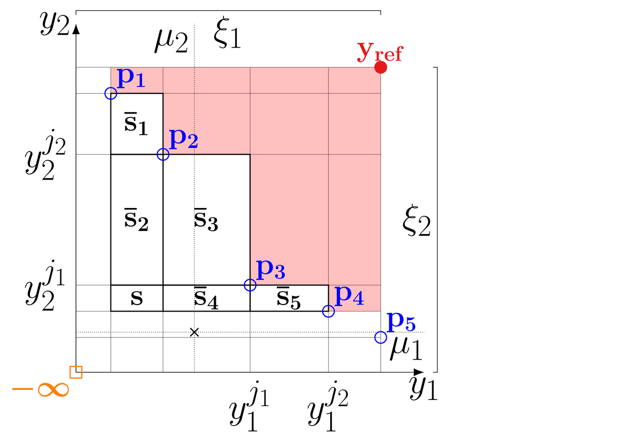
ALGORITHM:

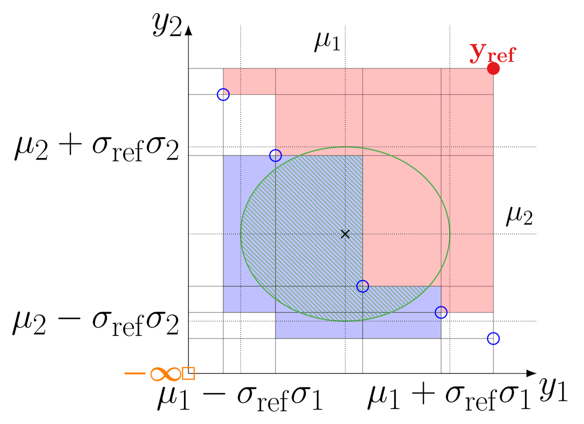
```
function OPTIMIZE(N_seq)                ▷ propose N_seq design points per iteration
  D ← INITIALCALCULATION(𝓧, S)           ▷ Random exploration
  while not STOP(D) do                   ▷ Evaluate stopping criterion
    M ← UPDATEMODELS(D)                  ▷ Train/refine surrogates
    D' ← D
    ν ← 0
    while ν < N_seq do
      x ← SUGGESTION(𝓧, M, D')           ▷ Propose single point
      ŷ, f̂ ← PREDICTION(M, x)            ▷ Evaluate surrogate
      D' ← D' ∪ {(x, ŷ, f̂)}
      ν ← ν + 1
    end while
    D_new ← CALCULATION(S, x(D'))        ▷ Evaluate simulation
    D ← D ∪ D_new                        ▷ Update data base
  end while
  return PARETO(D)                       ▷ Pareto optimal subset of acquired data
end function
```

3. Implementation

PYTHON CODE AVAILABLE ONLINE: github.com/RaoulHeese/adasamp-pareto

- Allows a sequential or a parallelized evaluation of the simulation.
- Handles regression and classification models with *scikit-learn* structure.
- Limited to probabilistic models with normal distributions (explicit formulas).
- Expected hypervolume improvement with novel *ellipsoid truncation method*.





4. Benchmark

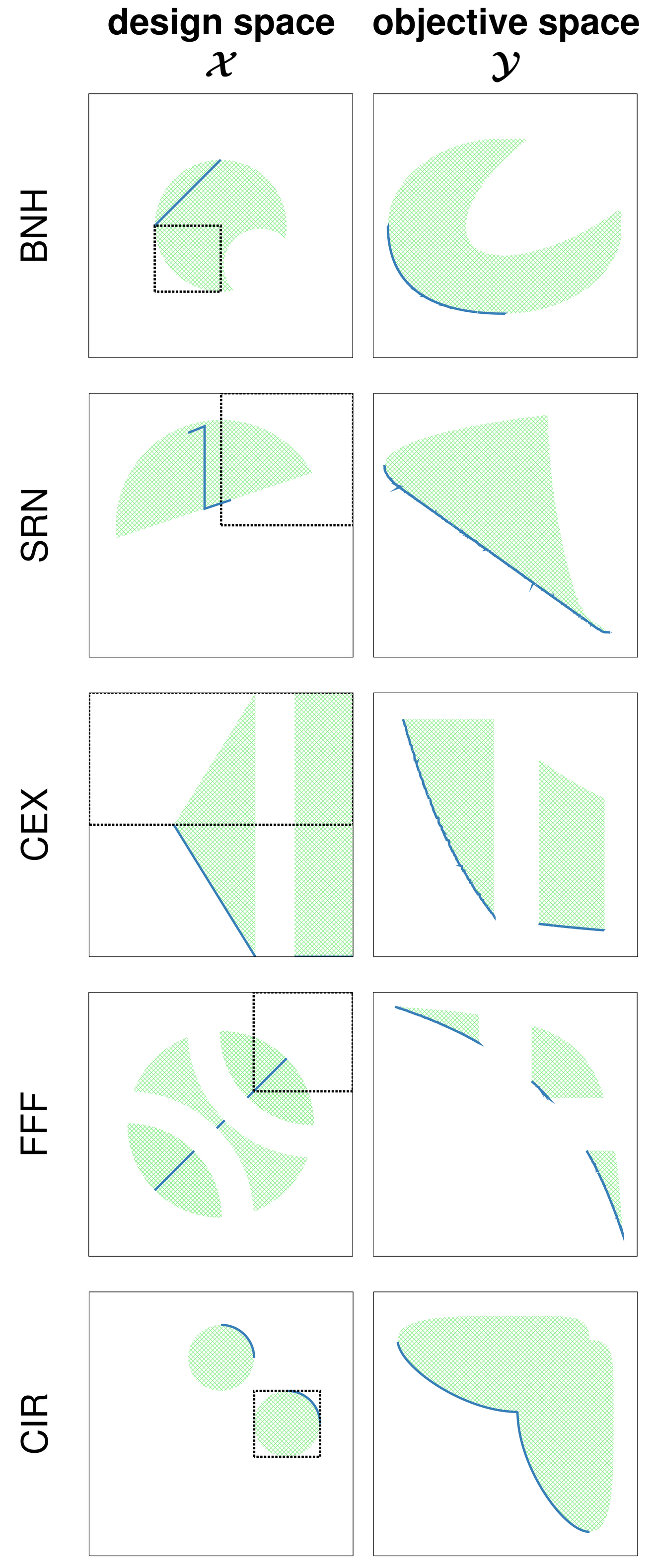
SETUP:

- We study 6 different benchmark problems (5 of which are from the literature).
- The feasibility f is determined from the mutual fulfillment of m problem-specific constraints $c_1(\mathbf{x}), \dots, c_m(\mathbf{x})$:

$$f(\mathbf{x}) = \begin{cases} \text{feasible} & \text{if } c_i(\mathbf{x}) \leq 0 \, \forall i = 1, \dots, m \\ \text{infeasible} & \text{else} \end{cases}$$

- To increase reproducibility, we select an initial sampling region in the design space.
- For each problem, we perform 50 independent optimization runs.
- As surrogate models we use GPR/BRR for $\hat{\mathbf{y}}$ and kSVM for $\hat{f}$.
- Our algorithm `adaptive-`N_{seq} competes against `nsgaii`.

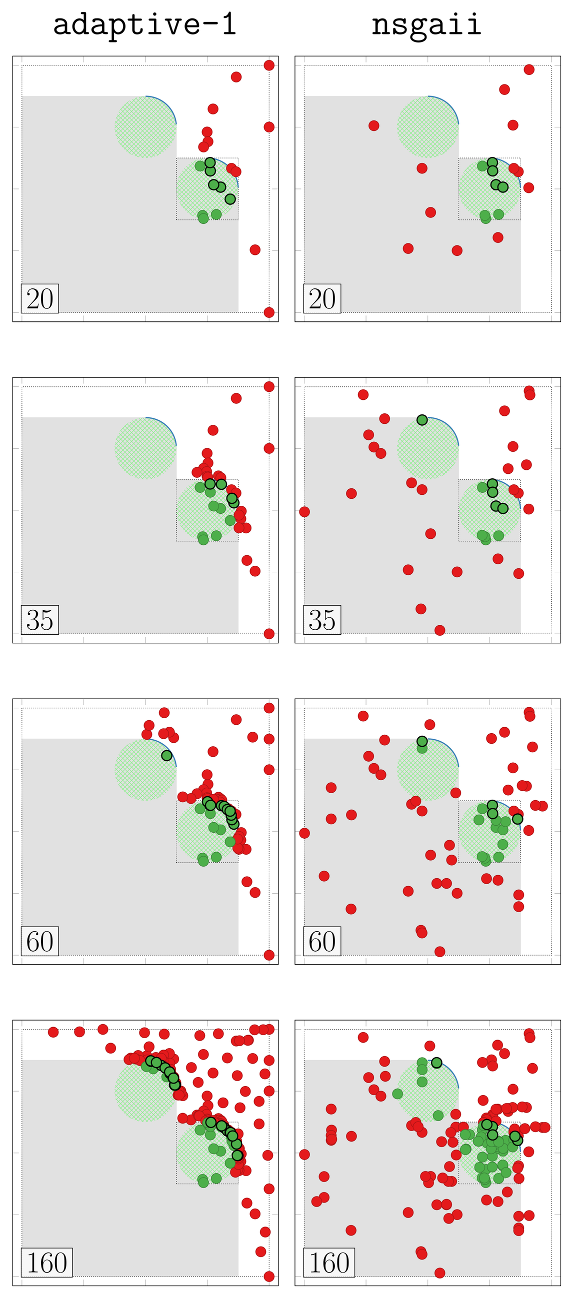
TEST PROBLEMS:



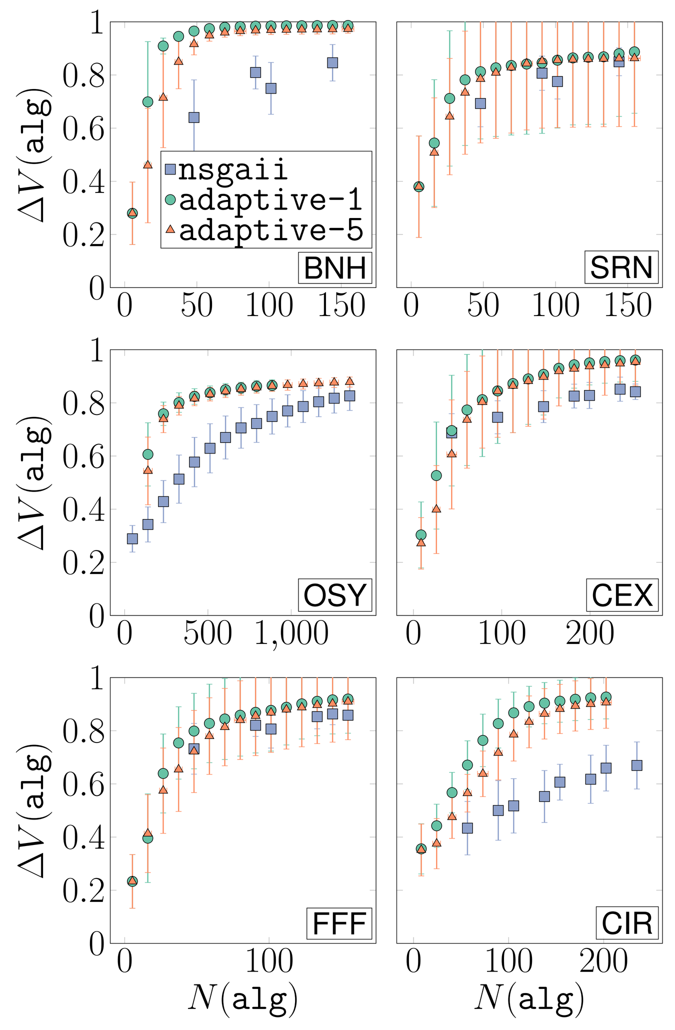
Problem name	$\dim(\boldsymbol{\mathcal{X}}) = d$	$\dim(\boldsymbol{\mathcal{Y}}) = n$	m constraints
BNH	2	2	2
SRN	2	2	2
OSY	6	2	6
CEX	2	2	4
FFF	2	2	3
CIR	2	2	1

5. Results

CIR SAMPLING OF $\boldsymbol{\mathcal{X}}$:



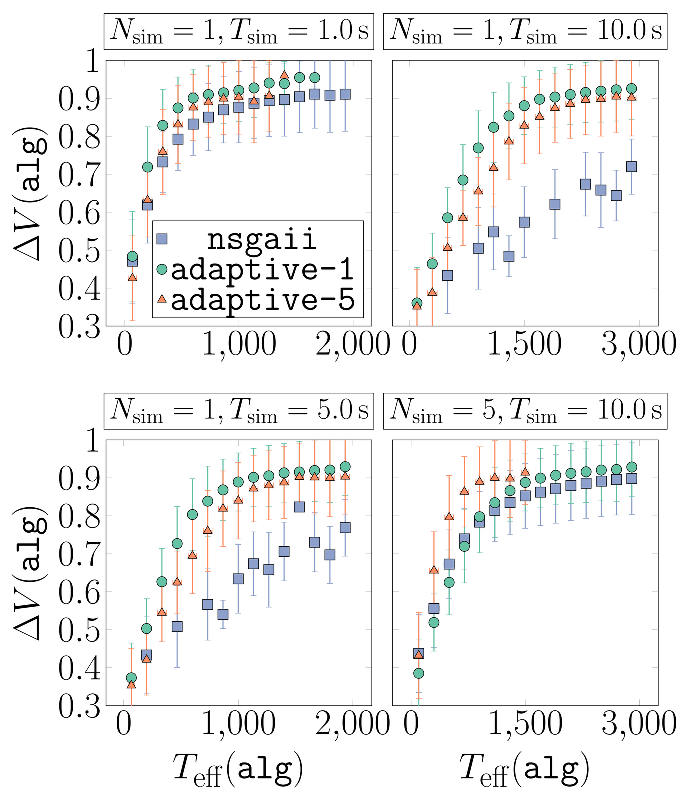
NUMBER OF EVALUATIONS:



- The relative total dominated volume with respect to a reference point $\mathbf{y}_{\mathbf{ref}} \in \boldsymbol{\mathcal{Y}}$ is defined as

$$\Delta V(\mathtt{alg}) \equiv \frac{V(\mathbf{D}(\mathtt{alg}), \mathbf{y}_{\mathbf{ref}})}{V_{\mathrm{true}(\mathbf{y}_{\mathbf{ref}})}} \in [0, 1].$$

CIR RUNTIME:



- If N_{sim} simulations can be evaluated in parallel during a constant runtime T_{sim} and the pure runtime of the algorithm is $T(\mathtt{alg})$, then the effective runtime after N_{iter} iterations reads

$$T_{\mathrm{eff}}(\mathtt{alg}) \equiv T_{\mathrm{sim}} \left\lceil \frac{N_{\mathrm{seq}}}{N_{\mathrm{sim}}} \right\rceil N_{\mathrm{iter}} + T(\mathtt{alg}).$$

- We iterate over different artificially chosen values of N_{sim} and T_{sim} for the benchmark.

SUMMARIZED RESULTS:

Problem name	Total number of evaluations $N^{\delta}(\mathtt{adaptive\text{-}1}, \delta v)$				Break-even simulation times $\tau(\mathtt{adaptive\text{-}1}, \mathtt{nsgaii}, \delta v)$			
	$\delta v = 0.80$	$\delta v = 0.85$	$\delta v = 0.90$	$\delta v = 0.95$	$\delta v = 0.80$	$\delta v = 0.85$	$\delta v = 0.90$	$\delta v = 0.95$
BNH	16.36 ± 2.54	18.82 ± 2.56	25.14 ± 4.89	38.30 ± 5.40	$(0.13 \pm 0.08)\,\mathrm{s}$	$(0.16 \pm 0.08)\,\mathrm{s}$	$(0.22 \pm 0.10)\,\mathrm{s}$	$(0.30 \pm 0.09)\,\mathrm{s}$
SRN	28.77 ± 23.26	31.77 ± 23.08	38.86 ± 22.54	62.40 ± 15.71	$(0.26 \pm 0.20)\,\mathrm{s}$	$(0.25 \pm 0.16)\,\mathrm{s}$	$(0.24 \pm 0.12)\,\mathrm{s}$	$(0.34 \pm 0.13)\,\mathrm{s}$
OSY	334.92 ± 121.41	556.11 ± 172.00	710.00 ± 112.00	> 750	$(1.49 \pm 1.34)\,\mathrm{s}$	$(2.91 \pm 2.37)\,\mathrm{s}$	$(2.83 \pm 1.56)\,\mathrm{s}$	–
CEX	57.82 ± 41.50	68.31 ± 42.04	84.81 ± 37.48	135.87 ± 37.88	$(0.65 \pm 0.76)\,\mathrm{s}$	$(0.52 \pm 0.52)\,\mathrm{s}$	$(0.45 \pm 0.33)\,\mathrm{s}$	$(0.50 \pm 0.28)\,\mathrm{s}$
FFF	46.29 ± 24.81	53.87 ± 27.50	70.53 ± 26.49	117.13 ± 20.78	$(1.09 \pm 1.17)\,\mathrm{s}$	$(0.97 \pm 1.19)\,\mathrm{s}$	$(1.00 \pm 1.02)\,\mathrm{s}$	$(2.41 \pm 1.05)\,\mathrm{s}$
CIR	74.04 ± 13.49	86.20 ± 14.22	108.09 ± 17.41	182.45 ± 15.37	$(0.46 \pm 0.23)\,\mathrm{s}$	$(0.44 \pm 0.21)\,\mathrm{s}$	$(0.42 \pm 0.22)\,\mathrm{s}$	$(0.48 \pm 0.16)\,\mathrm{s}$

- $N^{\delta}(\mathtt{adaptive\text{-}1}, \delta v)$: Number of evaluations to reach a certain relative total dominated volume $\Delta V(\mathtt{alg}) \geq \delta v \in [0, 1]$.
- $\tau(\mathtt{adaptive\text{-}1}, \mathtt{nsgaii}, \delta v)$: Break-even simulation times $T_{\mathrm{sim}} > \tau$, for which `adaptive-1` runs faster than `nsgaii` to reach a certain relative total dominated volume δv.

6. Summary

- We propose a **novel adaptive optimization algorithm** on the foundation of Bayes optimization, which allows us to solve black-box multi-objective optimization problems with black-box binary constraints.
- The **weight-based acquisition function** is intuitively understandable and can be tuned to the demands of the problems at hand.
- To speed up calculation time of the expected hypervolume improvement, we propose an **ellipsoid truncation method**.
- A **benchmark** has shown that our approach can compete with an evolutionary algorithm on a set of test problems with respect to the number of iterations and the calculation time.

7. Outlook

- Consider **noisy/uncertain simulations**, which would require an appropriate modification of the surrogate models.
- Consider **integer (and mixed-integer) design variables** which would allow us to solve integer programming problems and mixed-integer programming problems.
- Incorporate **additional non-binary constraints**.
- Use **analytically calculated gradients**, which could greatly improve the performance of the acquisition function optimization.
- Apply the method to more complex problems and real-world **applications**.