Simulation-based Bayesian Optimization

Roi Naveiro

CUNEF University

Cheap yet effective way of doing research

- Take a problem from the comp. sci. community
- Bring the statistical perspective / stats. methodology
- Why?
  - Very mathy for Comp. Sci. reviewers
  - Very fancy form Stat. reviewers

The CS problem

Find extreme of objective function

$$\arg\max_{x \in \mathcal{S}} f(x)$$

- Possibly discrete structured domain $\mathcal{S}$ (e.g. $\mathcal{S} = [k]^p$)
- No closed form for $f(x)$
- $f(x)$ is expensive to evaluate
- We can query at x and obtain a (possibly noisy) evaluation of $f(x)$
- Many applications: RNA design, molecular design…

Goal

- Get good estimates of global maximum, with **few objective function evaluations**
- A possible way to go: **Bayesian Optimization**

Bayesian Optimization (in a nutshell)

Imagine we have access to: $\mathcal{D}_{1:T} = \{x_t, y_t\}_{t=1}^{T}$ where

$$y_t \equiv y(x_t) \equiv f(x_t) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Problem: We have budget for H function evaluations. Using all available info, decide where to evaluate: $x_{T+1}, \ldots, x_{T+H}$

Solution: Locations maximizing Expected Utility[1] …too hard, go greedy: just decide next location!

1. Following Bayesian Decision Theory principles

Bayesian Optimization (in a nutshell)

To do this:

1. Create **probabilistic (Bayesian?) model** of response y given covariates x
2. Given prior knowledge about $f(x)$ and evidence coming from $\mathcal{D}_{1:T}$, summarise our **beliefs about result if we query at $\mathbf{x_{T+1}}$** through the **Posterior Predictive Distribution**

$$\pi[y_{T+1} | x_{T+1}, \mathcal{D}_{1:T}]$$

Bayesian Optimization (in a nutshell)

3. Find

$$x^*_{T+1} = \arg\max_{x_{T+1} \in \mathcal{S}} \mathbb{E}_{y_{T+1}|x_{T+1},\mathcal{D}_{1:T}}[u(y_{T+1}, x_{T+1})]$$

$$= \arg\max_{x_{T+1} \in \mathcal{S}} \int u(y_{T+1}, x_{T+1}) \pi(y_{T+1}|x_{T+1}, \mathcal{D}_{1:T}) dy_{T+1}$$

Bayesian Optimization - Difficulties

$\mathcal{S}$ could be a combinatorial search space (or even more complex!). We need:

1. Suitable models of response given covariates
2. Solve a combinatorial optimization in Step 3.

SBBO

We propose **Simulation Based Bayesian Optimization (SBBO)**: an approach to 2 that just requires **sampling from posterior predictive distribution**

$$\pi[y_{T+1} | x_{T+1}, \mathcal{D}_{1:T}]$$

allowing us to use a wide variaty of models which are suitable for combinatorial spaces

SBBO - Main idea

Convert EU maximization into a **simulation problem**.

The EU is:

$$\Psi(x) \equiv \int u(y, x) \cdot \pi(y|x, \mathcal{D}_{1:T}) dy$$

Given **non-negative** and **bounded** utility, recast EU maximization as a simulation from

$$g(x, y) \propto u(y, x) \cdot \pi(y|x, \mathcal{D}_{1:T})$$

NOTE: mode of marginal in x is x^*_{T+1} !

SBBO - Main idea

- We could simulate from $x, y \sim g(y, x)$ and find the mode in x... just limited for low dimensional x

SBBO - Main idea

- Alternative, Muller et. al. (2004): **auxiliary distribution**

$$g_H(x, y_1, \dots, y_H) \propto \prod_{h=1}^{H} u(y^h, x) \cdot \pi(y^h | x, \mathcal{D}_{1:T})$$

for positive integer H. Marginal in x

$$g_H(x) \propto \Psi(x)^H$$

SBBO - Main idea

Inhomogeneus MCMC simulation from $g_H(y, x)$ with increasing $H = H_n$ such that **stationary distribution** for fixed H is g_H, converges to uniform over set of expected utility maxmizers

SBBO - Gibbs

Let’s define

$$g_H(x, y_1, \dots, y_H) \propto \exp\left\{ \sum_{h=1}^{H} \log[u(y^h, x)] + \log[\pi(y^h | x, \mathcal{D}_{1:T}$$

SBBO - Gibbs

Recall $x \in [k]^p$ is a p-dimensional vector of k-levels categorical variables. Then:

$$g_H(x_q|\cdot) \propto \exp\left\{ \sum_{h=1}^{H} \log[u(y^h, x_q \cup x_{-q})] + \log[\pi(y^h|x_q \cup x_{-q}$$

Softmax over $\sum_{h=1}^{H} \log[u(y^h, x_q \cup x_{-q})] + \log[\pi(y^h|x_q \cup x_{-q}, \mathcal{D}_{1:T})]$ for every level x_q!

SBBO - Gibbs

Assume current state of chain is $x, y_1, \ldots, y_H$. Iterate

1. For $q = 1, \ldots, p$, sample $x_q \sim g_H(x_q|\cdot)$

2. For $h = 1, \ldots, H$, sample $y_h \sim g_H(y_h|\cdot)$, using a Metropolis-Hastings step.

3. Increase H according to chosen *"cooling" schedule*

SBBO - Gibbs

- Problem: recall that $g_H(x_q|\cdot)$

$$\texttt{softmax}\left(\sum_{h=1}^{H}\log[u(y^h, x_q \cup x_{-q})] + \log[\pi(y^h|x_q \cup x_{-q}, \mathcal{D}_{1:T})\right.$$

SBBO - Implementation details

- We run the previous algorithm increasing H until certain value.
- Last generated x is the new evaluation
- We propose several probabilistic models of response given discrete covariates for which we have sampling access to their **posterior predictive distribution (PPD)**

SBBO - Learners

- **BLr**: Bayesian second-order linear regression with heavy-tailed horseshoe prior
- **GPr**: GP with Tanimoto kernel
- NGBoost (Duan et. al., 2020): natural gradient boosting with two different base learners
  - **NGBdec**: Decision tree
  - **NGBlin**: Lasso linear regression
- To compare: Simulated Annealing (**SA**), Random Local Search (**RS**)

Experimental Results

- Initial dataset $\mathcal{D}_{1:T} = \{x_t, y_t\}_{t=1}^{T}$ with $T = 5$
- For $t = 1 : 500$ iterations:
  - Use SBBO to propose next sample: x_{t+1}. Utility function expected improvement: $u(y) = \max_y (y - y^*, 0)$
  - Evaluate true objective: $y_{t+1}(x_{t+1})$
  - Update dataset with (x_{t+1}, y_{t+1})
- Report best function value after t evaluations of true objective, averaged over 10 runs (plus/minus one standard error).

Contamination Problem

- Food supply with $d = 25$ stages that maybe contaminated
- Z_i denotes fraction of food contaminated at i-th stage
- At stage i, prevention effort (with cost c_i) can be made $(x_i = 1)$ decreasing contamination a random rate Γ_i
- If no prevention is taken $(x_i = 0)$, contamination spreads with random rate Λ_i

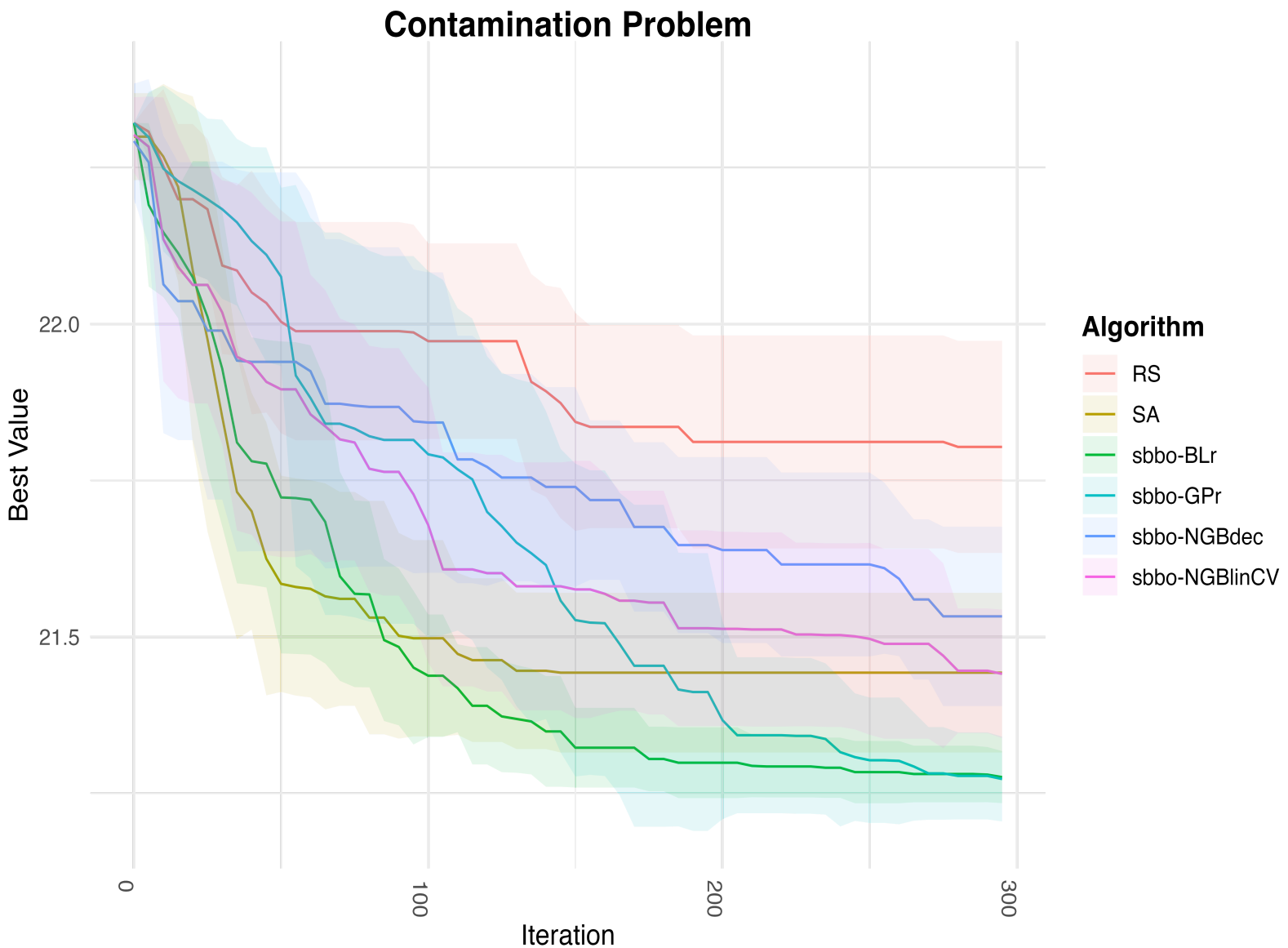
$$Z_i = \Lambda_i(1 - x_i)(1 - Z_{i-1}) + (1 - \Gamma_i x_i)Z_{i-1}$$

Contamination Problem - Goal

- Decide for each stage whether to intervene or not to minimize cost ($2^d = 2^{25}$ candidate solutions)
- Ensuring fraction of cont. food does not exceed U_i with probability at least $1 - \epsilon$
- Lagrangian relaxation

$$\arg\min_x \sum_{i=1}^{d} \left[c_i x_i + \frac{\rho}{T} \sum_{k=1}^{T} \left(1_{\{Z_{ik} > U_i\}} - (1 - \epsilon) \right) \right] + \lambda \|x\|_1$$

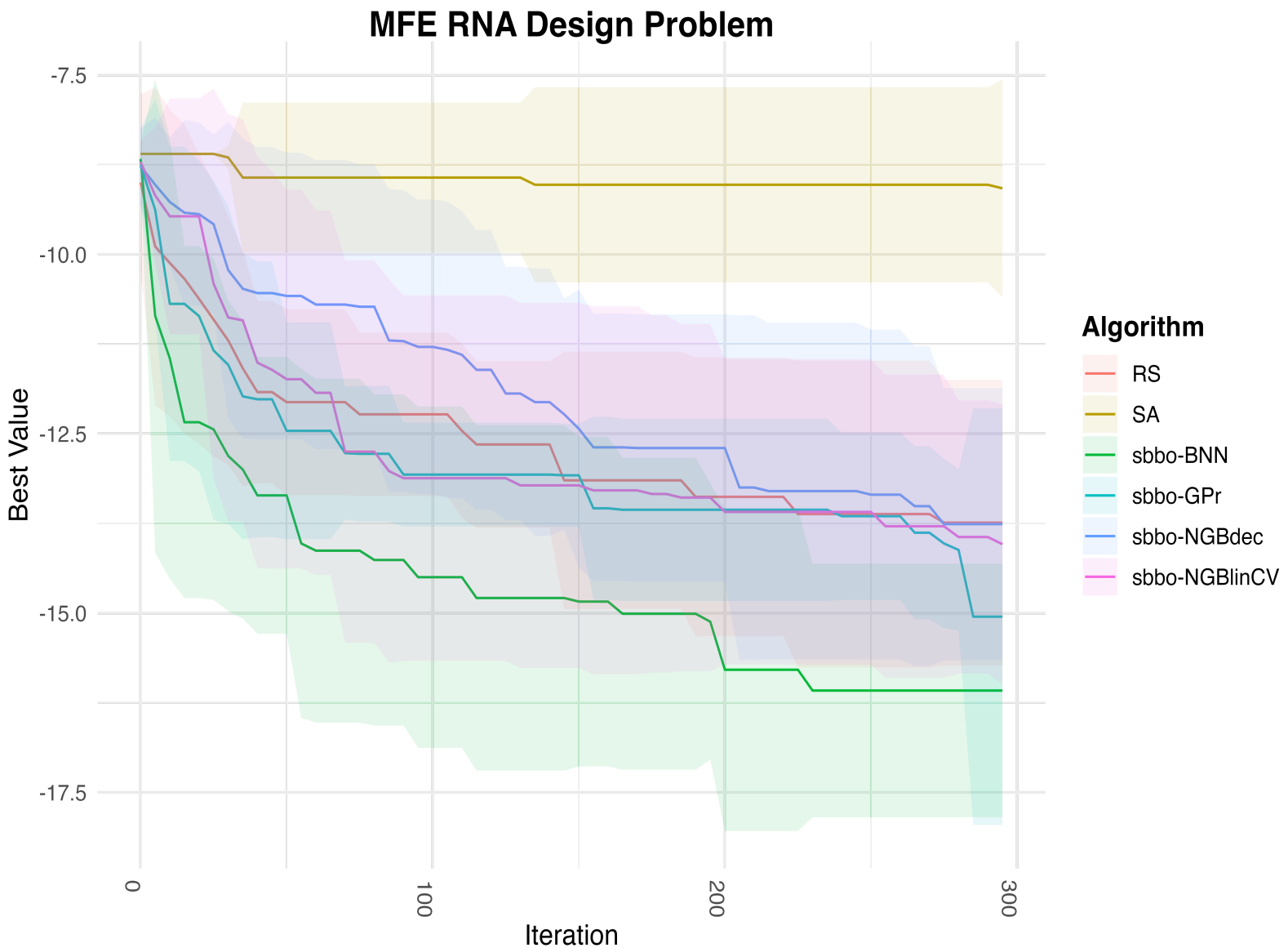
Results


Contamination Problem
Best Value
22.0
21.5
0
100
200
300
Iteration
Algorithm
RS
SA
sbbo-BLr
sbbo-GPr
sbbo-NGBdec
sbbo-NGBlinCV

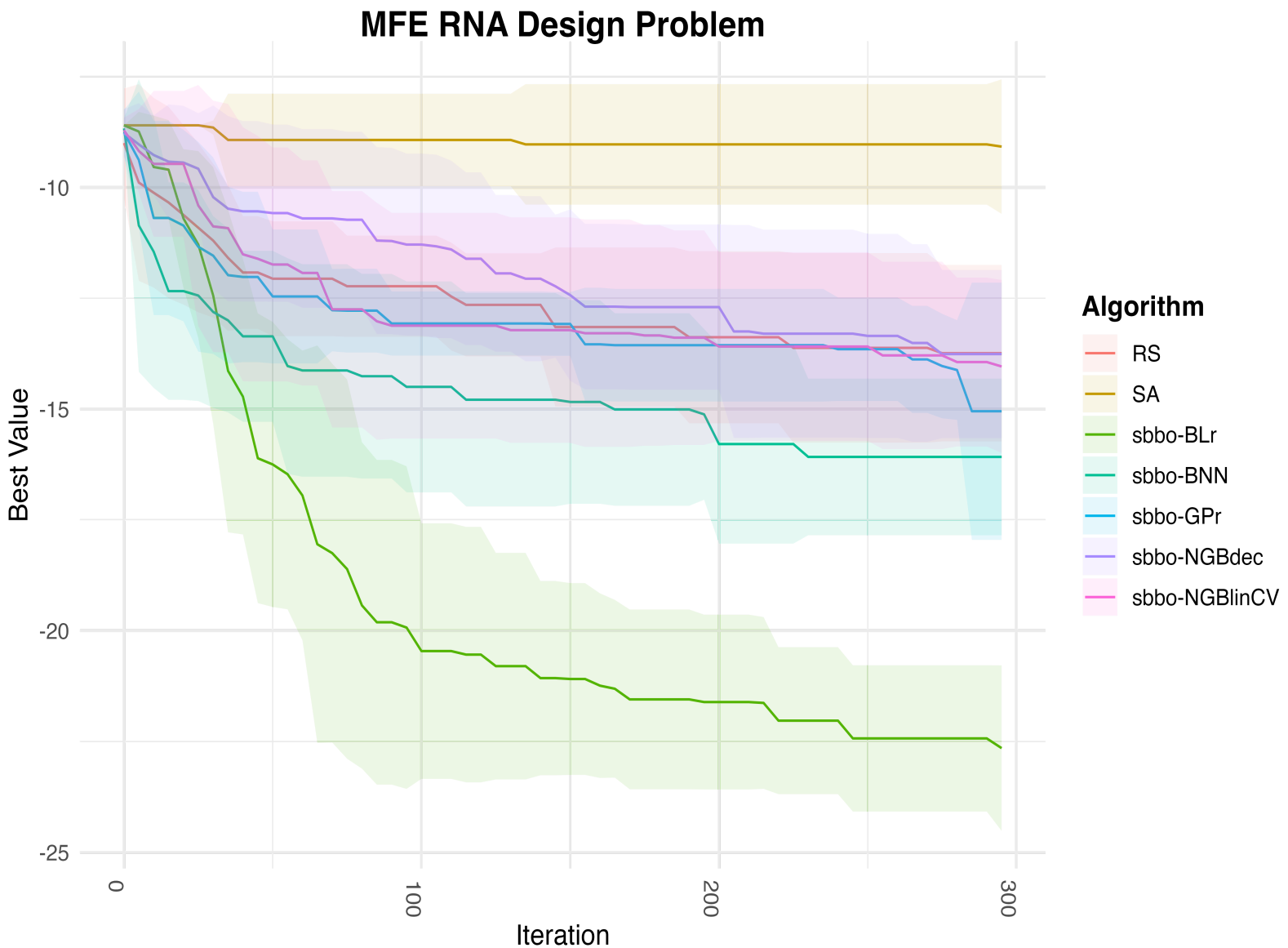
RNA design

- RNA sequence: string $A = a_1, a_2, \ldots, a_n$, with $a_i \in \{A, C, G, U\}$
- Secondary structure: ensemble of paring basis with minimum free energy (MFE)
- Estimate MFE via thermodynamic model and dynamic programming has time complexity $\mathcal{O}(n^3)$
- Searching for MFE structure is hard: * 4^n possibilites
- Use SBBO to find MFE RNA chain of lenght n with **few energy evaluations**
- Conv. Bayesian Neural Nets!

RNA design


MFE RNA Design Problem
-7.5
-10.0
-12.5
-15.0
-17.5
Best Value
0
100
200
300
Iteration
Algorithm
RS
SA
sbbo-BNN
sbbo-GPr
sbbo-NGBdec
sbbo-NGBlinCV

RNA design


MFE RNA Design Problem
Best Value
-10
-15
-20
-25
0
100
200
300
Iteration
Algorithm
RS
SA
sbbo-BLr
sbbo-BNN
sbbo-GPr
sbbo-NGBdec
sbbo-NGBlinCV

Conclusions

- SBBO allows to do BO with any surrogate model (as long as we can sample from its PPD)
- Opens the door to use many models (not used before in BO)
- SBBO can be used to optimize expected utilities not only in combinatorial search spaces, also in continuous and mixed ones!

Future Work

- Can we understand which algorithm will work best apriori?
- More models: Bayesian non-parametric models such as BART
- Go beyond the greedy approach…
  - Sequential decision problem
  - Generate multiple proposals at once (repulsion? S^3?)
- Exploit the idea of conjugate utilities

Thanks!

Code available at my github!

https://roinaveiro.github.io/

SBBO - Tanimoto GP

- Uncertainty on $f(x)$ modelled through Gaussian Process
- $x \in \{0, 1\}^p$. Kernel function:

$$k(x, x') = \frac{x \cdot x'}{\|x\|^2 + \|x'\|^2 - x \cdot x'}$$

- **PPD is Gaussian with certain mean and variance analytically available**

SBBO - Sparse Bayesian linear regression

$$y = \alpha_0 + \sum_j \alpha_j x_j + \sum_{i,j>i} \alpha_{ij} x_i x_j + \epsilon$$

$$\alpha_k | \beta_k, \tau, \sigma^2 \sim \mathcal{N}(0, \beta_k^2 \tau^2 \sigma^2)$$

$$\beta_k, \tau \sim C^+(0, 1)$$

$$P(\sigma^2) \propto \sigma^{-2}$$

- **PPD accesible through MCMC (Gibbs sampler)**

SBBO - NGBoost

Duan et. al. (2020):

- Output given covariates modelled through $y|x \sim P_\theta(x)$
- Where $\theta(x)$ are obtained through an additive combination of M base learners and an initial $\theta^{(0)}$

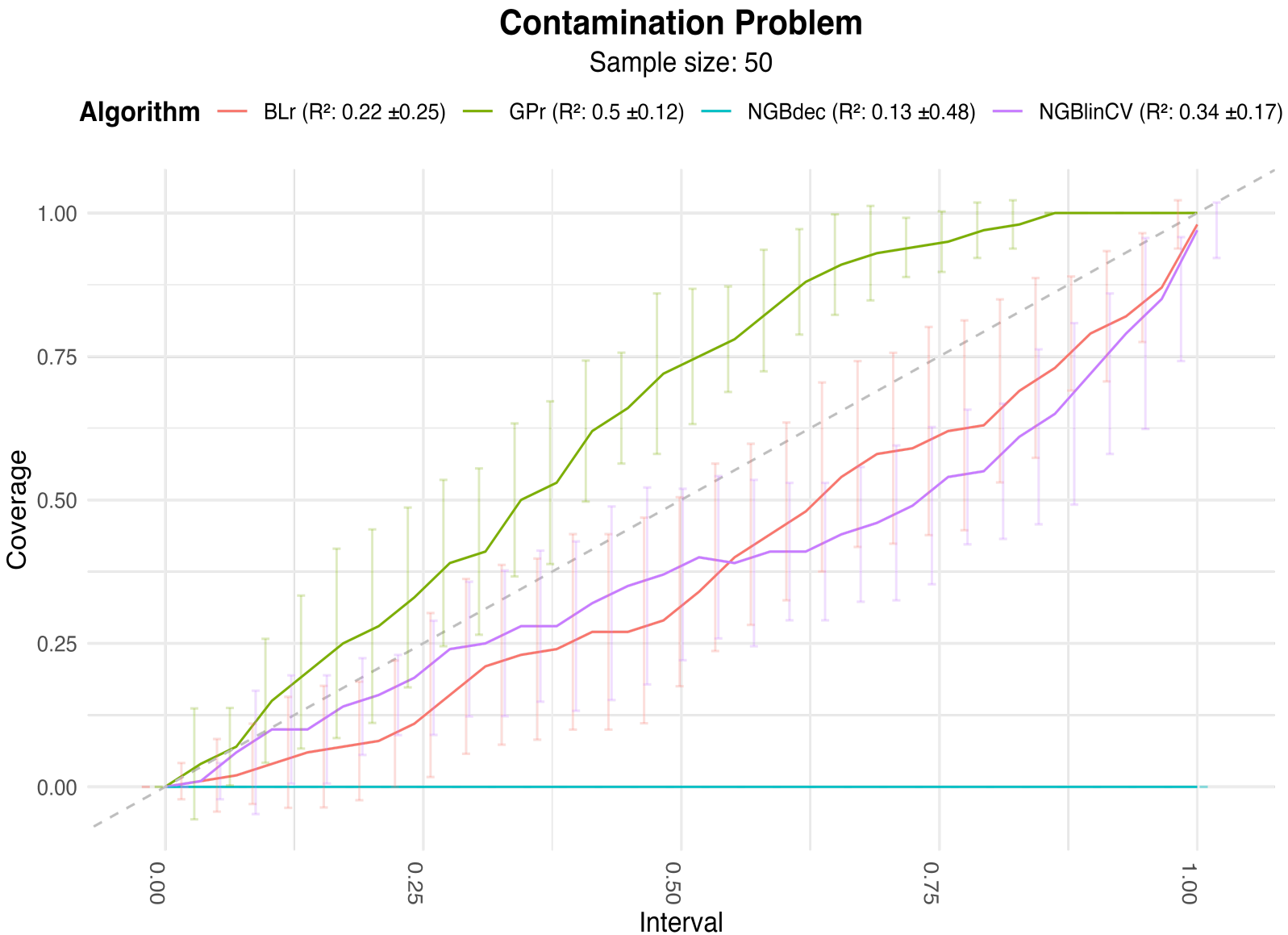
$$\theta = \theta^{(0)} - \eta \sum_{m=1}^{M} \rho^{(m)} \cdot f^{(m)}(x)$$

- Learners are trained to minimize a **proper scoring rule** using a refinement of **gradient boosting**

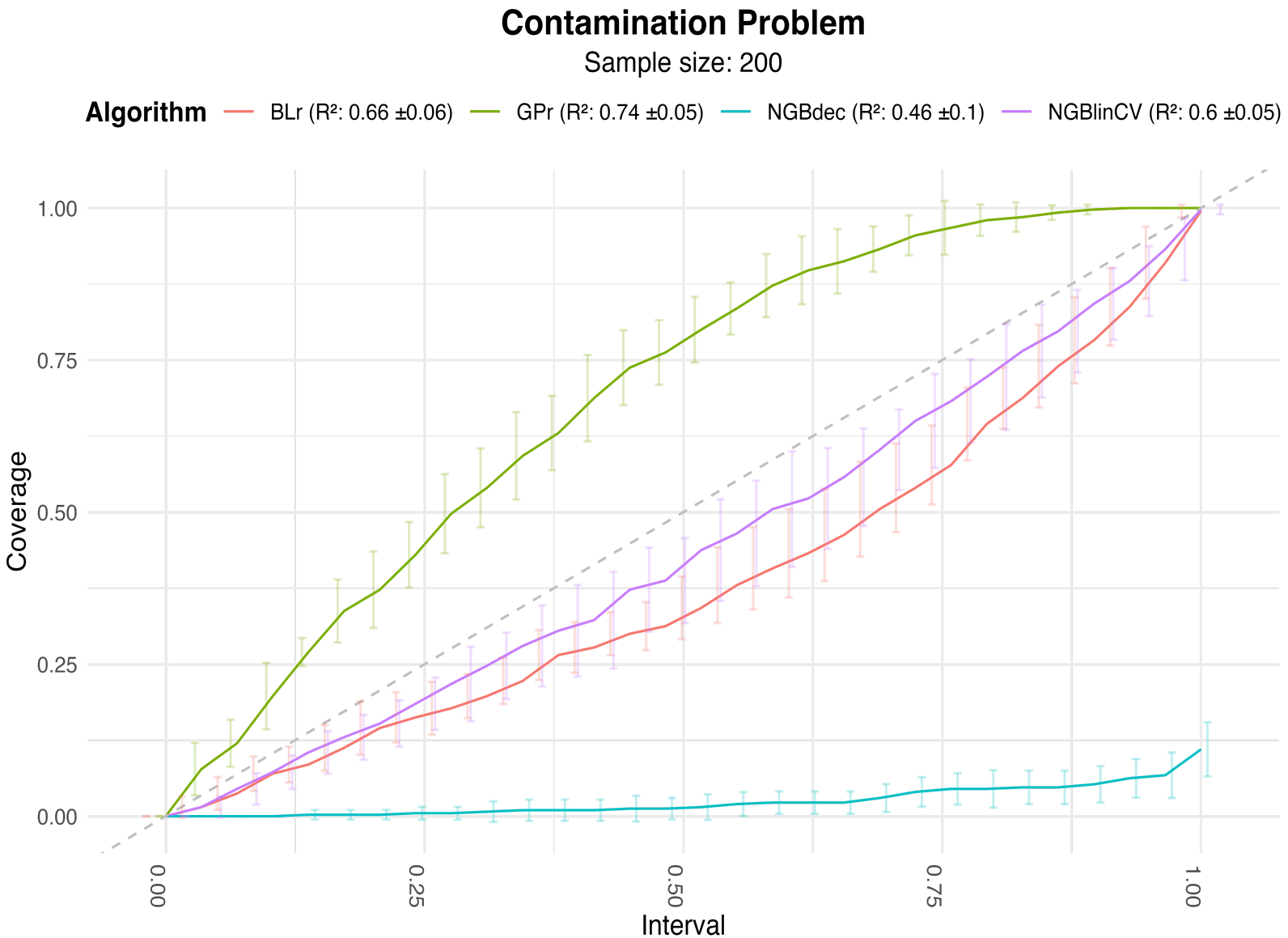
SBBO - NGBoost

- Any base learner can be used
- Base learners used: **shallow decision trees** and **linear regressions with lasso regularization**
- **PPD directly accesible**

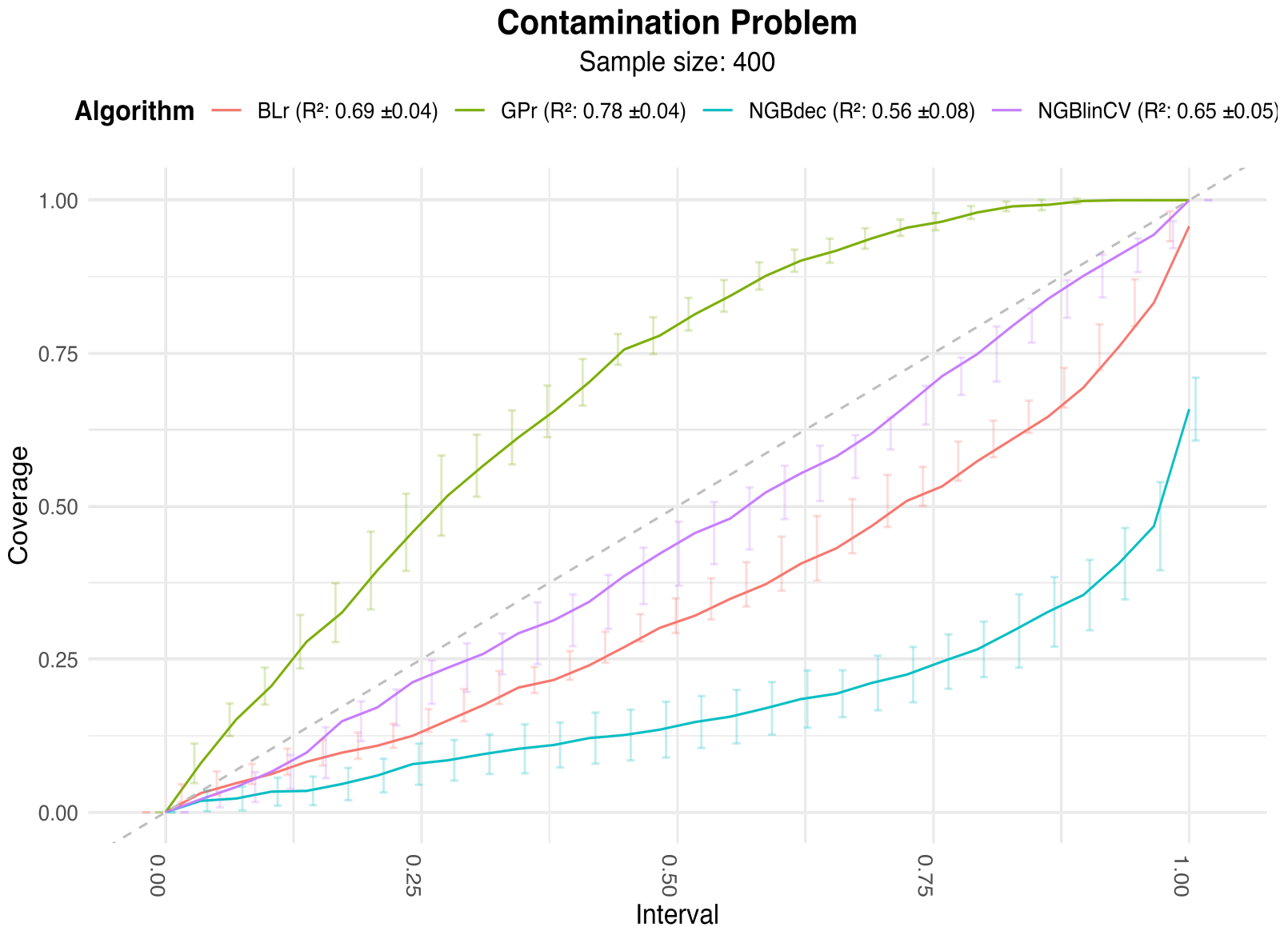
Results - Why?


Contamination Problem
Sample size: 50
Algorithm
BLr (R²: 0.22 ±0.25)
GPr (R²: 0.5 ±0.12)
NGBdec (R²: 0.13 ±0.48)
NGBlinCV (R²: 0.34 ±0.17)
Coverage
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Interval

Results - Why?


Contamination Problem
Sample size: 200
Algorithm
BLr (R²: 0.66 ±0.06)
GPr (R²: 0.74 ±0.05)
NGBdec (R²: 0.46 ±0.1)
NGBlinCV (R²: 0.6 ±0.05)
Coverage
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Interval

Results - Why?


Contamination Problem
Sample size: 400
Algorithm
BLr (R²: 0.69 ±0.04)
GPr (R²: 0.78 ±0.04)
NGBdec (R²: 0.56 ±0.08)
NGBlinCV (R²: 0.65 ±0.05)
Coverage
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Interval