Bayesian Optimization with Cost-varying Variable Subsets

Sebastian Shenghong Tay[1 2], Chuan Sheng Foo[2 3], Bryan Kian Hsiang Low[1] et. al.

[1]Department of Computer Science, National University of Singapore

[2]Institute for Infocomm Research (I2R), A*STAR, Singapore

[3]Centre for Frontier AI Research (CFAR), A*STAR, Singapore

Problem Formulation

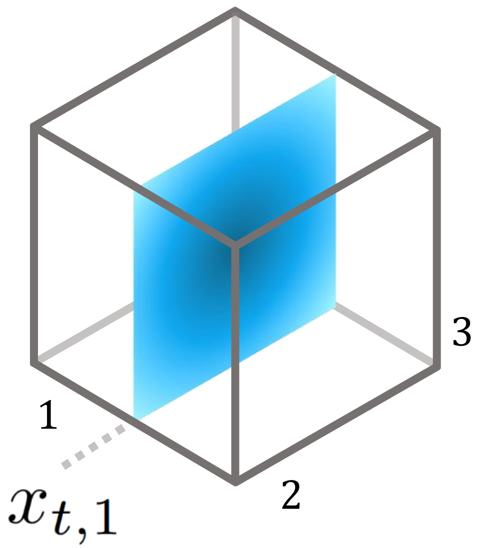
- Bayesian optimization (BO) is a powerful framework for the query-efficient optimization of costly-to-evaluate black-box objective functions. Standard BO assumes that all variables in a query $\mathbf{x}$ are controllable by the learner.
- However, in many real-world optimization problems, **some of the query variables may be subject to randomness** affecting their values.
- In some cases, the randomness affecting a specific variable can be **eliminated** (by allowing the learner to select its value), but **at a cost**.

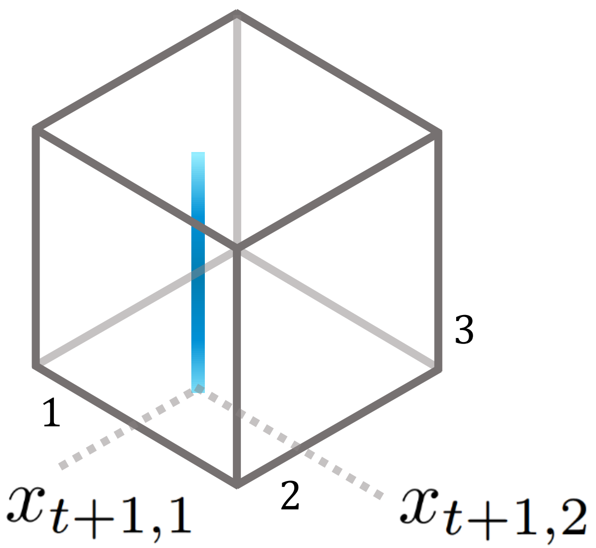
Problem Formulation

- Motivating example: In precision agriculture, consider a farm aiming to find the **optimal conditions for largest crop yield** where the query variables are a set of soil nutrient concentrations (e.g., calcium, potassium):
  - The farm may rely on the naturally-occurring quantities of these nutrients in the available soil, but these quantities will be **randomly sampled**;
  - alternatively, they may **control** some subset of these quantities (via manufactured soil and fertilizers) at a **higher cost**.
- General optimization problem: At each query iteration, the learner is faced with the challenges of
  - deciding **which variables to specify** (for more directed learning) vs. **which variables to allow to be randomly sampled** (to reduce incurred costs to avoid exceeding a given budget);
  - in addition to the usual optimization problem of **deciding the specified variables' values.**

Available **control sets**:

- Control set 1: choose 1st variable
- Control set 2: choose 1st and 2nd variables





1. Learner chooses **control set** $i_t = 1$
2. Learner chooses **value of 1st variable** $\mathbf{x}^{i_t} = (x_{t,1})$
3. Environment **randomly samples 2nd and 3rd variables** $\mathbf{X}^{-i_t} = (X_{t,2}, X_{t,3})$
4. Learner **pays cost** c_1 for choosing the 1st control set

1. Learner chooses **control set** $i_{t+1} = 2$
2. Learner chooses **value of 1st and 2nd variables** $\mathbf{x}^{i_{t+1}} = (x_{t+1,1}, x_{t+1,2})$
3. Environment **randomly samples 3rd variable** $\mathbf{X}^{-i_{t+1}} = (X_{t+1,3})$
4. Learner **pays cost** c_2 for choosing the 2nd control set

Optimization Objective

- The learner seeks the **optimal control set** and the **optimal partial query** associated with that control set, defined as

$$(i^*, \mathbf{x}^{i^*}) := \underset{(i,\mathbf{x}^i)\in[m]\times\mathcal{X}^i}{\operatorname{argmax}} \mathbb{E}\left[f([\mathbf{x}^i, \mathbf{X}^{-i}])\right]$$

- Every control set i has an associated **cost c_i**. The learner has a limited budget C, and **each query in a BO iteration expends c_i** depending on the control set chosen in that iteration.

Algorithm

Idea: use **cheap** (and likely more random) control sets for **exploration** and use **expensive** (and likely more deterministic) control sets for **exploitation.**

Algorithm 1 UCB-CVS

Input: GP with kernel k, budget C, control sets $\mathcal{I}$, costs $(c_i)_{i=1}^{m}$, $\boxed{\epsilon\text{-schedule } (\epsilon_t)_{t=1}^{\infty}}$

for iteration $t = 1$ **to** ∞ **do**

  $g_t := \max_{(i,\mathbf{x}^i)\in[m]\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$

  $\mathcal{S}_1 := \{i \in [m] \mid \max_{\mathbf{x}^i\in\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big] + \epsilon_t \geq g_t\}$

  $\mathcal{S}_2 := \{i \in \mathcal{S}_1 \mid c_i = \min_{j\in\mathcal{S}_1} c_j\}$

  $(i_t, \mathbf{x}^{i_t}) := \operatorname{argmax}_{(i,\mathbf{x}^i)\in\mathcal{S}_2\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$

  break if $C - \sum_{\tau=1}^{t-1} c_{i_\tau} < c_{i_t}$

  Observe $\mathbf{x}^{-i_t}$ drawn from $\mathbb{P}^{-i_t}$

  Observe $y_t := f(\mathbf{x}_t) + \xi_t$

  $\mathcal{D}_t := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^{t}$

end for

return $\mathcal{D}_t$

Idea achieved with decreasing ϵ-schedule

Algorithm

Idea: use **cheap** (and likely more random) control sets for **exploration** and use **expensive** (and likely more deterministic) control sets for **exploitation.**

Algorithm 1 UCB-CVS

Input: GP with kernel k, budget C, control sets $\mathcal{I}$, costs $(c_i)_{i=1}^{m}$, ϵ-schedule $(\epsilon_t)_{t=1}^{\infty}$

for iteration $t = 1$ **to** ∞ **do**

  $g_t := \max_{(i,\mathbf{x}^i)\in[m]\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$

  $\mathcal{S}_1 := \{i \in [m] \mid \max_{\mathbf{x}^i\in\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big] + \epsilon_t \geq g_t\}$

  $\mathcal{S}_2 := \{i \in \mathcal{S}_1 \mid c_i = \min_{j\in\mathcal{S}_1} c_j\}$

  $(i_t, \mathbf{x}^{i_t}) := \operatorname{argmax}_{(i,\mathbf{x}^i)\in\mathcal{S}_2\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$

  break if $C - \sum_{\tau=1}^{t-1} c_{i_\tau} < c_{i_t}$

  Observe $\mathbf{x}^{-i_t}$ drawn from $\mathbb{P}^{-i_t}$

  Observe $y_t := f(\mathbf{x}_t) + \xi_t$

  $\mathcal{D}_t := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^{t}$

end for

return $\mathcal{D}_t$

---

1. Compute the maximum expected upper confidence bound (UCB) value g_t across all control sets.

Algorithm

Idea: use **cheap** (and likely more random) control sets for **exploration** and use **expensive** (and likely more deterministic) control sets for **exploitation.**

Algorithm 1 UCB-CVS

Input: GP with kernel k, budget C, control sets $\mathcal{I}$, costs $(c_i)_{i=1}^m$, $\boxed{\epsilon\text{-schedule } (\epsilon_t)_{t=1}^{\infty}}$ **Decreasing ϵ-schedule**
for iteration $t = 1$ **to** ∞ **do**
  $g_t := \max_{(i,\mathbf{x}^i)\in[m]\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$
  $\mathcal{S}_1 := \{i \in [m] \mid \max_{\mathbf{x}^i\in\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big] + \epsilon_t \geq g_t\}$
  $\mathcal{S}_2 := \{i \in \mathcal{S}_1 \mid c_i = \min_{j\in\mathcal{S}_1} c_j\}$
  $(i_t, \mathbf{x}^{i_t}) := \operatorname{argmax}_{(i,\mathbf{x}^i)\in\mathcal{S}_2\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$
  break if $C - \sum_{\tau=1}^{t-1} c_{i_\tau} < c_{i_t}$
  Observe $\mathbf{x}^{-i_t}$ drawn from $\mathbb{P}^{-i_t}$
  Observe $y_t := f(\mathbf{x}_t) + \xi_t$
  $\mathcal{D}_t := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^t$
end for
return $\mathcal{D}_t$

1. Compute the maximum expected upper confidence bound (UCB) value g_t across all control sets.
2. Collect into the set $\mathcal{S}_1$ every control set i that, after an ϵ_t relaxation, attains g_t.

Algorithm

Idea: use **cheap** (and likely more random) control sets for **exploration** and use **expensive** (and likely more deterministic) control sets for **exploitation.**

Algorithm 1 UCB-CVS

Input: GP with kernel k, budget C, control sets $\mathcal{I}$, costs $(c_i)_{i=1}^{m}$, ϵ-schedule $(\epsilon_t)_{t=1}^{\infty}$
for iteration $t = 1$ **to** ∞ **do**
  $g_t := \max_{(i,\mathbf{x}^i)\in[m]\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$
  $\mathcal{S}_1 := \{i \in [m] \mid \max_{\mathbf{x}^i\in\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big] + \epsilon_t \geq g_t\}$
  $\mathcal{S}_2 := \{i \in \mathcal{S}_1 \mid c_i = \min_{j\in\mathcal{S}_1} c_j\}$
  $(i_t, \mathbf{x}^{i_t}) := \operatorname{argmax}_{(i,\mathbf{x}^i)\in\mathcal{S}_2\times\mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$
  break if $C - \sum_{\tau=1}^{t-1} c_{i_\tau} < c_{i_t}$
  Observe $\mathbf{x}^{-i_t}$ drawn from $\mathbb{P}^{-i_t}$
  Observe $y_t := f(\mathbf{x}_t) + \xi_t$
  $\mathcal{D}_t := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^{t}$
end for
return $\mathcal{D}_t$

1. Compute the maximum expected upper confidence bound (UCB) value g_t across all control sets.
2. Collect into the set $\mathcal{S}_1$ every control set i that, after an ϵ_t relaxation, attains g_t.
3. Retain only the cheapest control set(s).

Algorithm

Idea: use **cheap** (and likely more random) control sets for **exploration** and use **expensive** (and likely more deterministic) control sets for **exploitation.**

Algorithm 1 UCB-CVS

Input: GP with kernel k, budget C, control sets $\mathcal{I}$, costs $(c_i)_{i=1}^{m}$, ϵ-schedule $(\epsilon_t)_{t=1}^{\infty}$
for iteration $t = 1$ **to** ∞ **do**
  $g_t := \max_{(i,\mathbf{x}^i) \in [m] \times \mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$
  $\mathcal{S}_1 := \{i \in [m] \mid \max_{\mathbf{x}^i \in \mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big] + \epsilon_t \geq g_t\}$
  $\mathcal{S}_2 := \{i \in \mathcal{S}_1 \mid c_i = \min_{j \in \mathcal{S}_1} c_j\}$
  $(i_t, \mathbf{x}^{i_t}) := \operatorname{argmax}_{(i,\mathbf{x}^i) \in \mathcal{S}_2 \times \mathcal{X}^i} \mathbb{E}\big[u_{t-1}([\mathbf{x}^i, \mathbf{X}^{-i}])\big]$
  break if $C - \sum_{\tau=1}^{t-1} c_{i_\tau} < c_{i_t}$
  Observe $\mathbf{x}^{-i_t}$ drawn from $\mathbb{P}^{-i_t}$
  Observe $y_t := f(\mathbf{x}_t) + \xi_t$
  $\mathcal{D}_t := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^{t}$
end for
return $\mathcal{D}_t$

1. Compute the maximum expected upper confidence bound (UCB) value g_t across all control sets.
2. Collect into the set $\mathcal{S}_1$ every control set i that, after an ϵ_t relaxation, attains g_t.
3. Retain only the cheapest control set(s).
4. Among the control sets remaining, choose the one that attains the maximum expected UCB value and query the maximizing partial query.

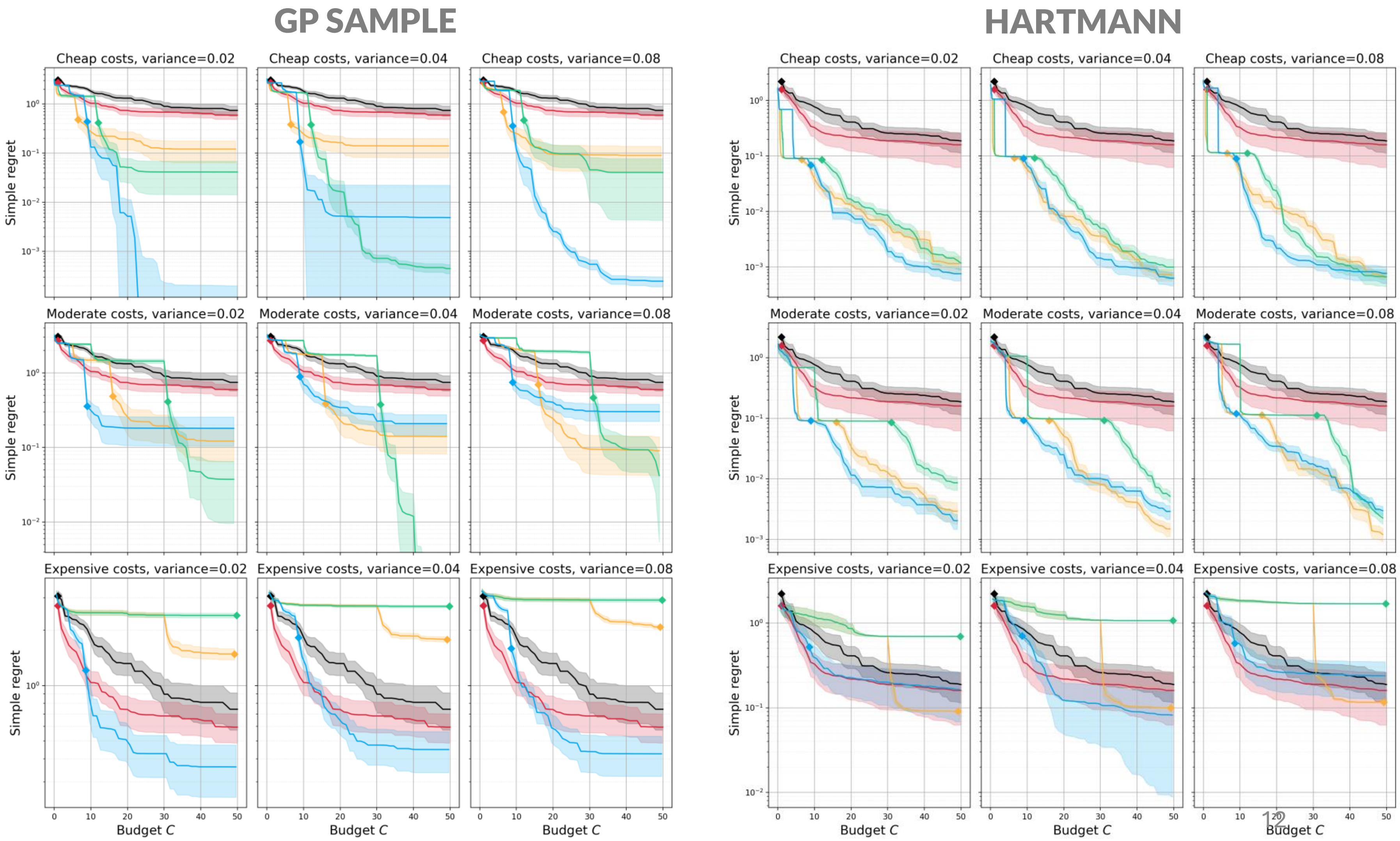
Theoretical Analysis

In the paper, we show:

1. Conditions on the ϵ-schedule under which UCB-CVS incurs **sublinear regret**.
2. How the **availability of cheaper control sets** and the **distributions** of the uncontrolled random variables affect regret.

Experiments

	TS-PSQ
Baselines	**UCB-PSQ**
	ETC-50
	ETC-100
Proposed algorithm	**ETC-Ada**



Diamond indicates average budget after which that algorithm only chooses the optimal control set

Experiments

Baselines

TS-PSQ

UCB-PSQ

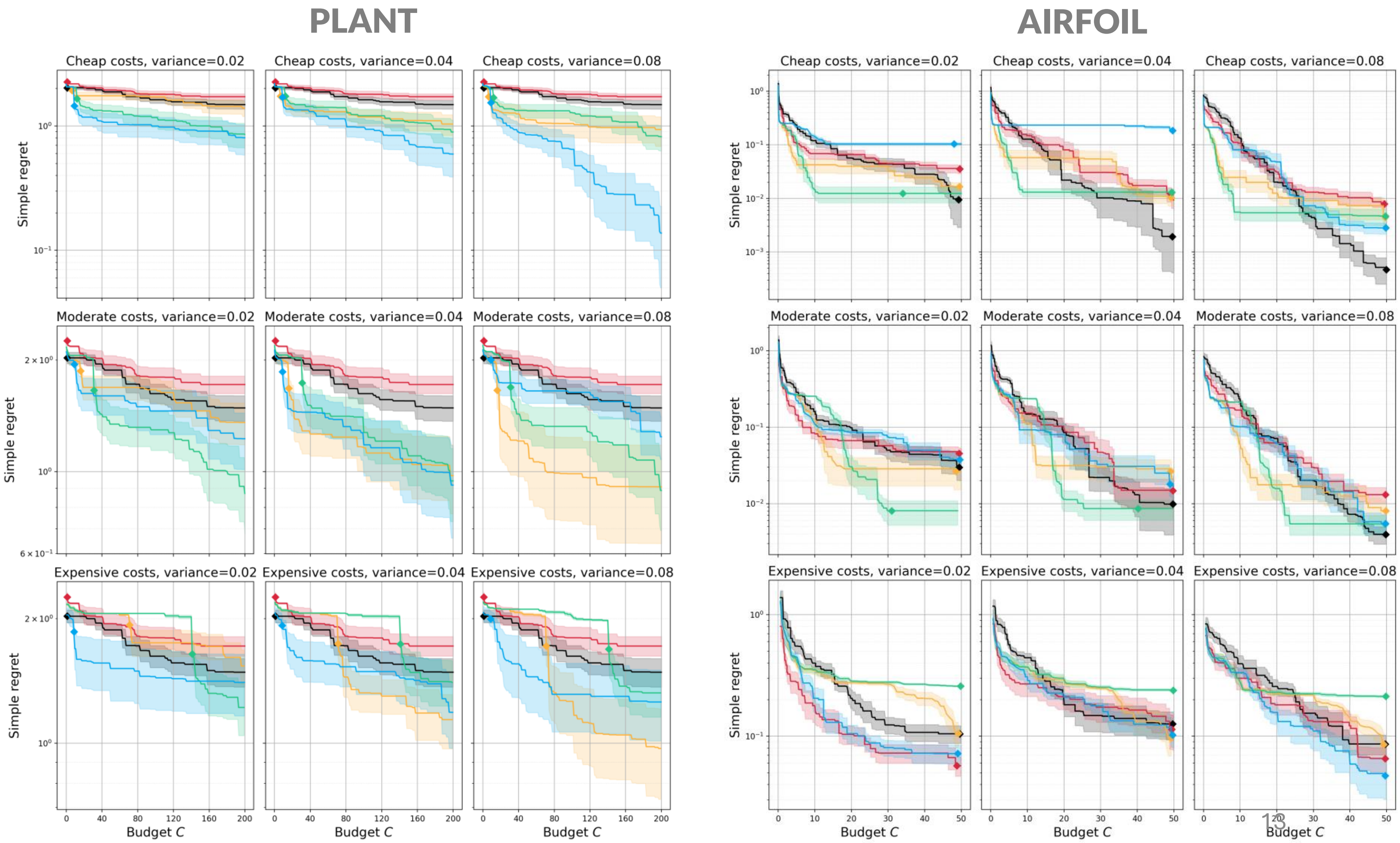
ETC-50

ETC-100

Proposed algorithm

ETC-Ada

Diamond indicates average budget after which that algorithm only chooses the optimal control set



Experiments Analysis

Experimental results suggest:

1. UCB-CVS variants outperform TS-PSQ and UCB-PSQ under **cheap/moderate costs** when the full query control set is available.
2. Cost-adaptive UCB-CVS (ETC-Ada) can maintain competitive performance **under expensive costs**.
3. Non-cost-adaptive TS-PSQ and UCB-PSQ perform relatively well when the **control sets are not subsets of each other**.
4. Increasing the variance of the probability distributions has **competing effects** on the simple regret.
5. Simple score-per-cost extensions of TS-PSQ, UCB-PSQ, and EI adapted for BOPSQ that simply divide acquisition score of a control set by its cost do not work well.

Summary

1. Introduce the BOCVS problem;
2. Solve the BOCVS problem by designing a novel UCB-based algorithm with a theoretical analysis of its properties;
3. Empirically evaluate the performance of our proposed algorithm against suitable baselines under several experimental settings with synthetic and real-world datasets.