Cost-aware Bayesian optimization via the Pandora's Box Gittins index

Qian Xie[1] **Raul Astudillo**[2] **Peter Frazier**[1] **Ziv Scully**[1] **Alexander Terenin**[1]
[1]Cornell University [2]Caltech

Abstract

Bayesian optimization is a technique for efficiently optimizing unknown functions in a black-box manner. To handle practical settings where gathering data requires use of finite resources, it is desirable to explicitly incorporate function evaluation costs into Bayesian optimization policies. To understand how to do so, we develop a previously-unexplored connection between cost-aware Bayesian optimization and the Pandora's Box problem, a decision problem from economics. The Pandora's Box problem admits a Bayesian-optimal solution based on an expression called the Gittins index, which can be reinterpreted as an acquisition function. We study the use of this acquisition function for cost-aware Bayesian optimization, and demonstrate empirically that it performs well, particularly in medium-high dimensions. We further show that this performance carries over to classical Bayesian optimization without explicit evaluation costs. Our work constitutes a first step towards integrating techniques from Gittins index theory into Bayesian optimization.

1 Introduction

Bayesian optimization is a framework for optimizing functions whose evaluation is time-consuming or expensive. It is widely used for hyperparameter tuning of machine learning algorithms [28], robot control [21], material design [36], and other areas. Bayesian optimization works by forming a probabilistic model for the objective function, and then chooses where to sample via an acquisition function that balances the explore-exploit trade-offs arising from uncertainty in this model.

We study *cost-aware* Bayesian optimization, where one must pay a cost to acquire another sample and this cost may vary with where the function is evaluated. Costs are an important factor in practical scenarios. For instance, in hyperparameter tuning using GPUs rented from a cloud provider, training a neural network for twice as many epochs carries twice the financial cost.

Despite its practical relevance, cost-aware Bayesian optimization is less-studied than standard Bayesian optimization, where budgets are framed in terms of the number of function evaluations and costs are not considered. Existing theoretically-principled cost-aware approaches [35, 17, 19, 3, 5] rely on multi-step lookahead computations that are computationally expensive and can be numerically brittle, limiting their applicability. Other approaches lack a theoretical foundation and risk having poor performance on certain problems. For example, one of the most popular cost-aware acquisition functions used in practice, expected improvement per unit cost [28], has recently been theoretically shown by Astudillo et al. [3] to perform arbitrarily-worse than the optimal policy. Thus, in the cost-aware setting, there is a need for theoretically-principled and computationally-straightforward acquisition functions with good empirical performance.

In this work, we develop such an approach. To do so, we introduce a novel link between cost-aware Bayesian optimization and a discrete-space decision problem from economics called the *Pandora's*

Preprint. Under review.

Code available at: HTTPS://GITHUB.COM/QIANJANEXIE/PANDORABAYESOPT.

Box problem [34, 6, 27, 24]. The Pandora's Box problem admits an explicit Bayesian-optimal solution. We show how this solution can be used to develop a novel acquisition function class for two cost-aware Bayesian optimization settings: (i) *expected budget-constrained* cost-aware Bayesian optimization, where there is a constraint on the expected cost of the samples taken, and (ii) *cost-per-sample* cost-aware Bayesian optimization where the costs incurred are subtracted from the objective function value. The resulting acquisition functions are closely connected to expected improvement variants, but incorporate costs in a different, non-multiplicative way.

We evaluate the proposed acquisition function, termed the *Pandora's Box Gittins index*, on a comprehensive set of experiments to understand its strengths and weaknesses. On both sufficiently-easy low-dimensional problems and too-difficult high-dimensional ones, performance is comparable to baselines. On medium-hard problems of moderate dimension, however, the proposed acquisition function tends to decisively outperform baselines, in the worst case matching their performance. Strikingly, this performance carries over to the classical setting with uniform costs. We also discuss limitations, including behavior on unimodal problems where lookahead-based baselines are strong.

The Pandora's Box Gittins index is a version of the *Gittins index* [13], a general framework for deriving optimal policies for a variety of bandit-like decision problems [33, 7, 15] which is widely-used in queueing theory and related areas [14, 1, 26]. Our work thus opens a novel angle of attack for designing acquisition functions specialized to specific practical settings of interest.

Contributions. In this work, we (i) connect the Pandora's Box problem with a variant of cost-aware Bayesian optimization over a discrete search space. Using this connection, we (ii) explore the use of Gittins indices, which are Bayesian-optimal for the Pandora's Box problem, as an acquisition function for general cost-aware Bayesian optimization where data is incorporated via the posterior distribution. We (iii) demonstrate the resulting acquisition function has strong empirical performance on a variety of problems of moderate-to-high dimension, including the heterogeneous-cost problems it was designed for, as well as classical non-cost-aware problems.

2 Cost-aware Bayesian optimization

In black-box optimization, we are interested in finding the global optimum of an unknown (potentially stochastic) function $f : X \to \mathbb{R}$ defined on some compact domain, using pointwise function evaluations of f at locations $x_t \in X$ that we select sequentially. We are interested in policies achieving a small *simple regret* [11, Sec. 10.1]

$$\mathbb{E} \sup_{x \in X} f(x) - \mathbb{E} \max_{t=1,..,T} f(x_t) \tag{1}$$

where the expectation is taken with respect to all randomness in the function and procedure. Obtaining a new function evaluation at a point x carries a *cost* $c(x) \in \mathbb{R}_+$. We consider two settings that integrate costs into the problem in different ways:

(a) In the *expected budget-constrained* setting, there is a budget $B \in \mathbb{R}_+$, and the algorithm is not allowed to exceed this budget in expectation.

(b) In the *cost-per-sample* setting, at each time the algorithm must choose whether to pay a cost and obtain a new function evaluation, or to stop and return some previously-observed point. In this setting, we add the total sum of costs at termination time to the regret.

Note that the cost function $c : X \to \mathbb{R}_+$ can be constant, which we term *uniform costs*. In this case, (a) reduces to standard black-box optimization with a finite time horizon, and (b) reduces to variant of stopping-aware Bayesian optimization. These are not the only possible settings: one can also consider almost-sure budget constraints and other variants. Since we are interested primarily in the role of costs rather than stopping times in this work, we mostly work with budget constraints throughout this paper, but will use the cost-per-sample setting as a conceptual framework with which to study the budget-constrained setting.

2.1 Probabilistic models and acquisition functions

Bayesian optimization algorithms for solving various black-box optimization problems work by (i) building a *probabilistic model* of f—that is, a probability distribution which quantifies what is known about f given the data points $(x_t, y_t)_{t=1}^T$ seen so far, where $y_t = f(x_t)$ are previous function evaluations, then (ii) using the model and its uncertainty to decide where to evaluate the unknown function next. For an introduction, see Frazier [10] and Garnett [11]. Following standard practice, we work with Gaussian process models [25]. Let $f \mid y_1, .., y_T$ be the respective posterior distribution.

To decide where to evaluate f next, one uses the model to define a (potentially random) *acquisition function* $\alpha^{(t)} : X \to \mathbb{R}$, which quantifies how promising a particular location is given what is known so far. We then evaluate f at

$$x_{t+1} = \operatorname*{arg\,max}_{x \in X} \alpha^{(t)}(x), \tag{2}$$

obtaining an additional data point that is used to reduce uncertainty and further improve the model.

2.2 Expected improvement per unit cost

The most popular cost-aware acquisition function is *expected improvement per unit cost (EIPC)* [28], defined via

$$\alpha_{\mathrm{EIPC}}^{(t)}(x) = \frac{\mathrm{EI}_{f \mid y_1,..,y_t}(x; \max_{\tau \in [t]} y_\tau)}{c(x)} \qquad \mathrm{EI}_\psi(x; y) = \mathbb{E} \max(0, \psi(x) - y) \tag{3}$$

where we have written $\alpha_{\mathrm{EIPC}}^{(t)}(\cdot)$ in terms of the general *expected improvement function* EI_ψ, defined with respect to some random function $\psi : X \to \mathbb{R}$, and comparator point y. With this notation, EIPC can be interpreted as the ratio between the costs and expected improvement with respect to the current posterior, using the best point seen so far as the comparator.

In the uniform-cost case, where $c(x) = C \in \mathbb{R}_+$ for all x, this acquisition function reduces to the classical *expected improvement (EI)* acqusition function, namely $\alpha_{\mathrm{EI}}^{(t)}(x) = \mathrm{EI}_{f \mid y_1,..,y_t}(x; \max_{\tau \in [t]} y_\tau)$. In turn, expected improvement can be derived by considering the setup where the unknown function f is randomly sampled from the model's prior. If we imagine that the optimization process continues for one more time step, and stops after that, one can show that maximizing expected improvement is the optimal strategy in expectation.

Since EIPC reduces to expected improvement in the uniform-cost case where $c(x) = C$, it follows that it chooses the same points whether $C = 0.0001$ or $C = 1\,000\,000$. This is somewhat peculiar: one might expect that a cost-aware acquisition function should be more risk-averse if costs are high, and vice versa if they are low. Thus, EIPC is perhaps best suited to settings where heterogeneity is the main factor at play. However, even there, Astudillo et al. [3] show there exist reasonable problems where EIPC performs arbitrarily worse than the optimal policy in an approximation-ratio sense.

In spite of this rather negative theoretical outlook, EIPC has been shown to work well on many practical problems, is computationally efficient and reliable, and is in widespread use. We therefore ask: *can one develop a technically-principled and computationally-straightforward alternative with at-least-comparable empirical performance?*

3 The Pandora's Box Gittins index for Bayesian optimization

To develop a cost-aware acquisition function, we study a simplified decision problem that captures key difficulties of the main problem but is tractable enough to yield analytic insights. An analogous strategy is used classically to derive expected improvement, by exactly solving a simplified one-step decision problem. We study a different simplified decision problem, which can also be solved exactly, but where the simplification is spatial rather than temporal in nature. Specifically, we connect Bayesian optimization with the *Pandora's Box* problem from economics. To do so, we describe Pandora's Box in Section 3.1 and its solution in Section 3.2, showing along the way how these ideas can be reinterpreted from the view of Bayesian optimization. We illustrate this in Figure 1. Then, in Section 3.3, we use Pandora's Box to derive a novel class of cost-aware acquisition functions.

3.1 The Pandora's Box problem

The *Pandora's Box* problem [34, 13] is a sequential decision-making problem. It begins with a finite set of boxes, which we collect into a set and label $X = \{1, .., N\}$. Each box has a *hidden*

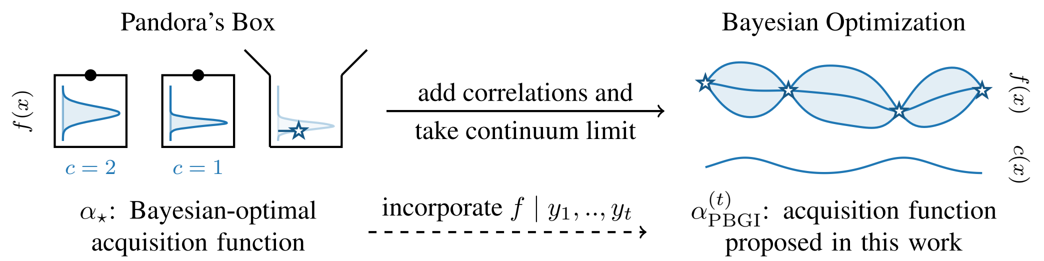
Pandora's Box
Bayesian Optimization
$f(x)$
$c = 2$
$c = 1$
add correlations and
take continuum limit
$f(x)$
$c(x)$
$\alpha_\star$: Bayesian-optimal acquisition function
incorporate $f \mid y_1, .., y_t$
$\alpha^{(t)}_{\text{PBGI}}$: acquisition function proposed in this work

Figure 1: An illustration of this work's key idea. We view cost-aware Bayesian optimization as an extension of the Pandora's Box problem, and derive the cost-aware acquisition function $\alpha^{(t)}_{\text{PBGI}}$ by incorporating the posterior into the Bayesian-optimal Pandora's Box acquisition function $\alpha_\star$.

reward, denoted by $f(x)$, and an *inspection cost*, denoted by $c(x)$. The rewards are given by mutually independent random variables whose distributions are known and vary between different boxes.

The decision-making process starts with a set of closed boxes, and proceeds in discrete time steps. At time t, one can choose to do one of two things:

1. *Open a box x_t.* This incurs cost c, but reveals the exact value $f(x_t)$ of the reward inside the box, which is drawn using the box's respective distribution.
2. *Stop opening new boxes, and take the reward from the best opened box.* This ends the decision-making process, and yields a terminal reward equal to the maximum value among the boxes opened so far, with the convention that if no boxes are opened the maximum is $-\infty$.

The policy's goal is to maximize the expected net reward, which is the reward of the best open box minus the total cost of all boxes opened so far, and is written

$$\mathbb{E} \max_{1 \leq t \leq T} f(x_t) - \mathbb{E} \sum_{t=1}^{T} c(x_t) \tag{4}$$

where T is a random variable that denotes the number of opened boxes, indicating that the policy terminates at time $T + 1$.

If we subtract the objective (4) from $\mathbb{E} \sup_{x \in X} f(x)$, which is constant with respect to the policy, we obtain the sum of the simple regret objective defined in Section 2 and total costs. *The Pandora's Box problem is therefore equivalent to a special case of cost-aware black-box optimization*, specifically the cost-per-sample variant of Section 2, where (a) the domain X is a finite set, and (b) the objective function f is random, with independent $f(x)$ and $f(x')$ for $x \neq x'$. We will return to this point in the sequel, but first study the Pandora's Box problem in more detail.

3.2 Optimally solving Pandora's Box

The Pandora's Box problem gives rise to an explore-exploit tradeoff: a policy must balance the opportunity gained from learning whether a box contains a large reward with the cost of opening the box to find this out. Since the reward distributions are known, this tradeoff is captured within a Markov decision process (MDP). By general MDP theory, there exists an optimal policy describing which box, if any, one should open for a given configuration—we call such a policy *Bayesian-optimal*.

This MDP can be solved explicitly, with a remarkably simple solution, first derived by Weitzman [34] We start by associating with each box $x \in X$ a number $\alpha_\star(x)$ known as the *Gittins index* [13]. Define

$$\alpha_\star(x) = g \qquad \text{where } g \text{ solves} \qquad \text{EI}_f(x; g) = c(x) \tag{5}$$

where $\text{EI}_f(x; y)$, previously defined in (3) of Section 2, is the *expected improvement of x relative to y*—the same expression which appeared inside the expected improvement acquisition function variants $\alpha^{(t)}_{\text{EI}}$ and $\alpha^{(t)}_{\text{EIPC}}$. Note that, unlike in those cases, $\alpha_\star$ is *not time-dependent* due to the lack of correlations or conditioning. By convexity, (5) admits a unique solution for every value of $c(x)$.

To understand what $\alpha_\star(x)$ represents, consider a single closed box x, and suppose there is a second, open box with reward f^*. Is opening box x better than taking the reward f^* from the open box? This

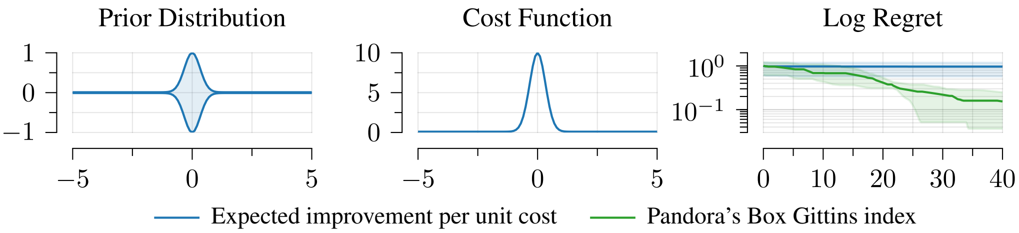

Figure 2: A Bayesian optimization problem with heterogeneous costs on which EIPC has poor performance, inspired by Astudillo et al. [3], Section A. The domain is $X = [-500, 500]$, which we visualize on the subinterval $[-5, 5]$. Left: illustration of the non-uniform prior variance, which is given by a Matérn-5/2 kernel scaled by a narrow bump function. Center: the cost function, which is a narrow bump-shaped function. Right: regret curves for EIPC and PBGI.

amounts to whether the expected improvement from opening x balances out the opening cost $c(x)$: one can show that opening x is better if and only if $\mathrm{EI}_f(x; f^*) > c(x)$. The value $\alpha_\star(x)$ tells us *how large does the alternative reward f^* need to be, for stopping to be at least as good as opening box x*—a kind of *fair price* which makes different boxes directly comparable to one another.

If we decide which box to open via the aforementioned fair prices, we obtain the *Gittins index policy*, which proceeds as follows. At each time t, let $f_t^* = \max_{1 \leq \tau \leq t} f(x_\tau)$ be the maximum reward among all open boxes, and let x_t^* be the box of maximum Gittins index value $\alpha_\star(x)$ among unopened boxes, with ties broken arbitrarily. With this notation:

- If $\alpha_\star(x_t^*) > f_t^*$, the policy opens box x^*.
- If $\alpha_\star(x_t^*) \leq f_t^*$, the policy stops and receives terminal reward f_t^*.

It turns out that opening boxes according to the order determined by their fair price, in the sense above, is not only a good idea, but is outright Bayesian-optimal. We state this formally as follows.

Proposition 1 (Weitzman [34])**.** *Let X be a finite set, let $f : X \to \mathbb{R}$ be a finite-mean random function for which $f(x)$ is independent of $f(x')$ for $x \neq x'$, and let $c : X \to \mathbb{R}_+$, without loss of generality, be deterministic. Then, for the cost-per-sample problem, the policy defined by maximizing the Gittins index acquisition function $\alpha_\star$ with its associated stopping rule is Bayesian-optimal.*

In the language of Bayesian optimization, this means that not only is there an explicit Bayesian-optimal policy for the Pandora's Box setting, this policy *takes the form of maximizing an acquisition function*. This gives an explicit solution for the cost-per-sample setting, thereby showing Pandora's Box fits our original goal of finding a simplified decision problem that sheds insights on cost-aware Bayesian optimization. For an alternative proof, see Kleinberg et al. [18], Theorem 1. Using Lagrangian relaxation, one can show the obtained solution carries over to the expected budget-constrained setting.

Proposition 2 (Corollary of Aminian et al. [2], Theorem 1)**.** *Consider the expected budget-constrained problem, with the assumptions of Proposition 1. Assume $B > \inf_{x \in X} c(x)$. Then there exists a $\lambda > 0$ such that the policy defined by maximizing the Gittins index acquisition function $\alpha_\star(\cdot)$ with its associated stopping rule, defined using costs $\lambda c(x)$, is Bayesian-optimal.*

Further technical details on both statements are given in Appendix B—including a proof of Proposition 2, which we provide for completeness due to minor technical differences between our setup and that of Aminian et al. [2]. The optimal λ depends on the budget constraint B implicitly via a convex optimization problem. In budget-constrained problems, we therefore view λ as a hyperparameter, which controls the degree to which the algorithm is risk-averse vs. risk-seeking—precisely what we argued was missing from EIPC in Section 2.

3.3 An acquisition function class for cost-aware Bayesian optimization

To adapt $\alpha_\star$ to the Bayesian optimization setting, we need to handle two differences: (i) X need not be discrete, and (ii) a general probabilistic model is used for f. Since Proposition 1 ostensibly requires $f(x)$ to be independent of $f(x')$ for all $x \neq x'$, the key question is *how to incorporate data*

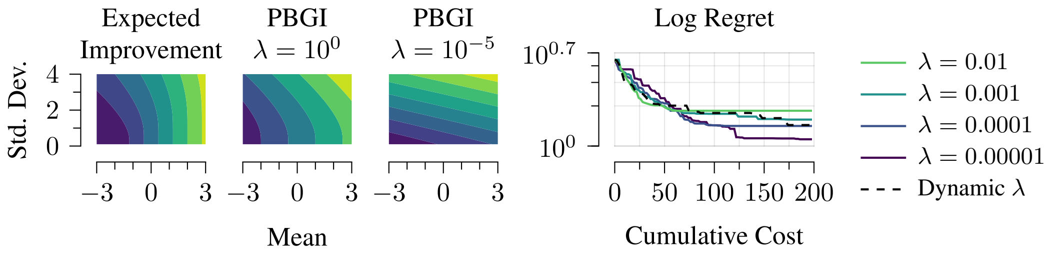

Figure 3: Left: contour plots showing how EI (left) and PBGI (center-left, center-right) depend on the posterior mean and standard deviation at a given point (lighter colors indicate higher values). We see that PBGI values high standard deviation more than EI. Right: PBGI performance across values of λ. We see that large λ-values decrease regret sooner, but eventually lose out to smaller λ-values.

and spatial correlations into $\alpha_\star$. We propose to do so in the simplest and most obvious way: namely, at each time t, we plug the posterior distribution $f \mid y_1, .., y_t$ in place of f. This yields three variants, depending on the precise cost-aware setting one is interested in:

1. *Budget-constrained*: define *Pandora's Box Gittins index* acquisition function

$$\alpha_{\text{PBGI}}(x) = g \qquad \text{where } g \text{ solves} \qquad \text{EI}_{f \mid y_1, .., y_t}(x; g) = \lambda c(x) \tag{6}$$

   and λ is a hyperparameter that should be tuned to match the evaluation budget.

2. *Cost-per-sample*: we can directly apply α_{PBGI} in this setting as well, but now λ is instead interpreted as unit-conversion factor which ensures costs and rewards have the same units, and the Pandora's Box stopping rule is used for deciding when to terminate the optimization procedure and return the best observed value.

3. *Cost-aware anytime*—that is, without an explicit budget or cost-based stopping rule: define the *Pandora's Box Gittins index with dynamic decay* $\alpha_{\text{PBGI-D}}(x)$ analogously to $\alpha_{\text{PBGI}}(x)$, but where λ is replaced with a time-dependent λ_t set using the Pandora's Box stopping rule. Specifically, first set λ_1 to an initial value, then at all times τ where the Pandora's Box stopping rule triggers, set $\lambda_{\tau+1} = \beta\lambda_\tau$, where $\beta < 1$ is the decay parameter, otherwise set $\lambda_{t+1} = \lambda_t$. The advantage of this is that one can potentially avoid tuning λ.

To understand this acquisition function class, one can think of it via the following approximation: for the general cost-aware Bayesian optimization problem, we (a) correctly incorporate observed data into the prior to obtain the posterior, but then (b) pick new samples according to the rule that would have been Bayesian-optimal if the posterior had no correlations. Said differently, α_{PBGI} arises from exactly solving a simplified dynamic program, where the simplification is of a spatial nature, rather than the usual temporal lookahead. One can therefore expect this acquisition function to work best in situations where correlations are not the decisive factor for determining performance.

In what problems does this happen? In stationary kernels, correlations encode local dependence. Therefore, one can expect α_{PBGI} to be approximately-optimal in settings where the key decisions involves choosing between different far-away data points. One can intuitively expect this to occur more often in high-dimensional problems, where the volume of the search space is large and most points are far away from each other. We will examine this point empirically in the sequel.

Computation. To compute α_{PBGI} efficiently, note that $y \mapsto \text{EI}_\psi(x; y)$ is monotone. As a result, the value g can be computed efficiently via bisection search. In Appendix B.3, we show that (i) its gradient can be computed straightforwardly via an explicit analytical expression without any additional optimization, and (ii) the resulting computational costs are much closer to those of expected improvement than those of expensive multi-step-lookahead-based approaches.

Qualitative behavior and comparisons. Compared to non-cost-aware acquisition functions such as expected improvement, the Pandora's Box Gittins index can act more risk-aversely if costs are large or more risk-seekingly if costs are small. In heterogeneous-cost budget-constrained settings, this tradeoff is mediated by λ, and the obtained decisions can differ significantly from those of widely-used baselines such as expected improvement per unit cost. In particular, PBGI can make qualitatively different decisions on problems where there is a high-variance point with a large cost,

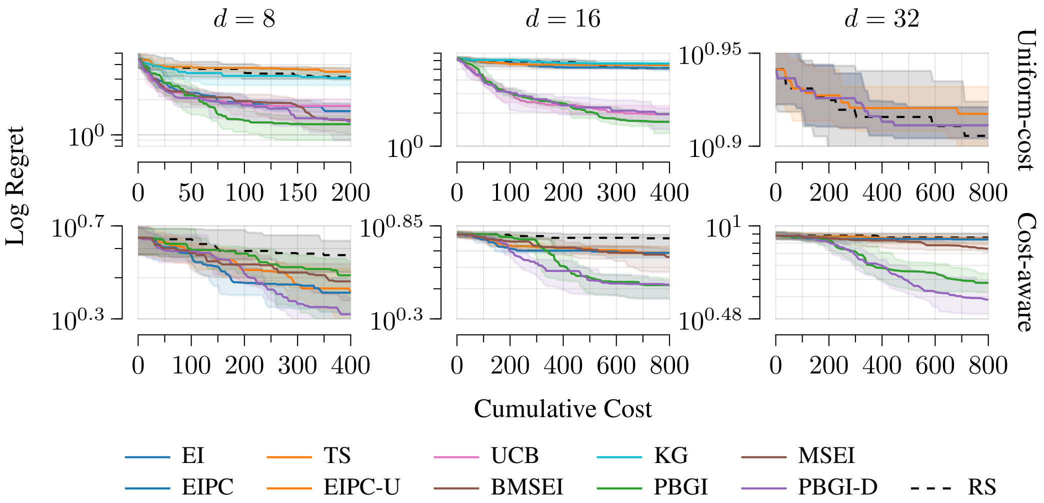

Figure 4: Bayesian regret curves, shown using medians, as well as quartiles to indicate experiment variability. We see in the cost-aware setting that both PBGI variants exhibit comparable performance to baselines for $d = 8$, and decisively outperform baselines in $d = 16$ and $d = 32$. This behavior is roughly-mirrored in the uniform-cost setting, with two notable distinctions: (a) UCB also exhibits strong performance for $d = 16$ matching PBGI and PBGI-D, and (b) all methods perform comparably to random search for $d = 32$ under uniform costs.

among a set of many low-variance low-cost points. In Figure 2, we adapt the construction of Astudillo et al. [3], Section A into a one-dimensional Bayesian optimization problem with a non-stationary prior, and observe that EIPC indeed has substantially worse performance than PBGI.

The PBGI acquisition functions depends on $f \mid y_1, .., y_t$ through its mean and standard deviation at each point. We plot this in Figure 3. This shows for large λ that PBGI can resemble expected improvement, whereas for small λ it is nearly linear, similar to the upper confidence bound (UCB) acquisition function whose dependence is exactly linear. For small λ, one can thus view PBGI as giving a way to automatically tune UCB's confidence parameter in a careful way depending on $c(x)$.

4 Experiments

We now empirically evaluate the Gittins-index-based acquisition function on cost-aware problems. We also evaluate on the same problems with a spatially-constant cost function, a setting we term *uniform costs*—this facilitates comparisons with classical, non-cost-aware baselines. In both cases, mirroring practical settings, we work with a deterministic, algorithm-independent evaluation budget.

We implement all methods in BoTorch [4] using Matérn Gaussian processes with smoothness $\nu = 5/2$ and length scale $\kappa = 0.1$. To ensure that our results are not sensitive to these and other hyperparameter choices, all experiments were repeated with alternatives given in Appendix C, including larger κ. Each experiment was repeated for 16 seeds to assess variability. Experimental details are in Appendix C.

We evaluate both PBGI variants of Section 3.3, namely α_{PBGI} with $\lambda = 0.0001$, and $\alpha_{\text{PBGI-D}}$ with $\beta = 0.5$. To ensure that performance differences are not primarily due to tuning, we deliberately use the same λ-and-β-values on all problems, even though per-problem tuning could be advantageous.

For cost-aware problems, we compare with *expected improvement per unit cost (EIPC)* and *budgeted multi-step expected improvement (BMSEI)*, which was proposed by Astudillo et al. [3] and has state-of-the-art cost-aware performance. To understand what happens if we simply ignore the cost function, we also compare against ordinary (that is, uniform-cost) *expected improvement (EIPC-U)*. For uniform-cost (that is, non-cost-aware) problems, we compare with *expected improvement (EI)*, *Thompson sampling (TS)*, *upper confidence bound (UCB)* [29], *knowledge gradient (KG)* [9], and *multi-step expected improvement (MSEI)* [17]. These were chosen because they are standard, and because acquisition function optimization succeeds for them on our problems, reducing confounding.

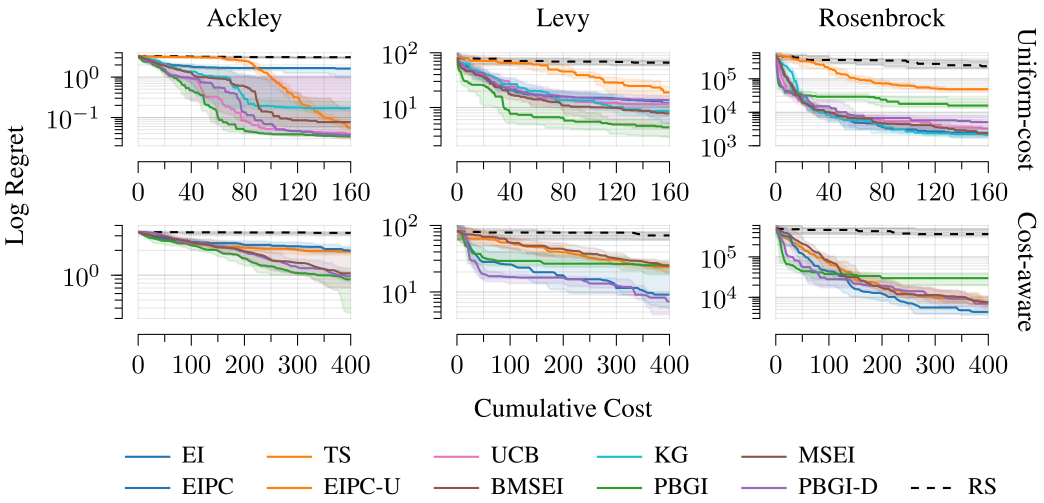

Figure 5: Synthetic benchmark regret curves, shown using medians, as well as quartiles to assess variability. All objective functions are defined with dimension $d = 16$. We see in the cost-aware setting that PBGI and PBGI-D perform strongest on the heavily-multimodal Ackley function, matching the non-myopic BMSEI baseline. On the Levy function, PBGI-D instead matches, and for some cost budgets outperforms, the EIPC baseline, significantly outperforming BMSEI. In contrast, on the unimodal Rosenbrock function, PBGI and PBGI-D only perform best for small cost budgets, and eventually end up matching BMSEI, and for large cost budgets, end up outperformed by EIPC, whose myopic behavior takes advantage of unimodality. Uniform-cost results are similar: PBGI performs well on Ackley and Levy, but is outperformed by most baselines and PBGI-D on Rosenbrock.

4.1 Bayesian regret

For our first experiment, we examine how well the proposed acquisition functions perform on random functions sampled from the prior. To quantify the effect of problem difficulty, we vary the dimension of the domain $X = [0, 1]^d$, and consider $d \in \{8, 16, 32\}$. Results, in terms of empirical regret curves and their associated quantiles, are shown in Figure 4. Additional results for $d = 4$, which show both PBGI variants and all baselines achieving similar performance, are in Appendix C.

In the low-dimensional case of $d = 8$, most uniform-cost and cost-aware approaches achieve similar performance. Once we increase dimension to $d = 16$, we see bigger differences: here, both PBGI variants achieve a modest improvement compared to expected improvement per unit cost. Strikingly, both PBGI variants are also competitive in the uniform-cost setting—in spite of being designed for cost-aware problems. This can be explained via the curse of dimensionality: as dimension increases, the problem begins to look more like the uncorrelated Pandora's Box problem where using Gittins index is Bayesian-optimal. Eventually, however, the problem becomes too difficult for meaningful progress to be made without substantial computational resources for optimizing acquisition functions, as seen for the uniform-cost problem with $d = 32$, where no method outperforms random search.

4.2 Synthetic benchmarks

Next, we consider standard synthetic global optimization benchmark functions. To represent a variety of geometric properties, we examine the *Ackley*, *Levy*, and *Rosenbrock* functions. A visualizaton of the two-dimensional versions of these functions is given in Appendix A.

Figure 5 presents results for $d = 16$. Additional results for $d = 4, 8$ showing that PBGI and all baselines perform similar, are in Appendix C. We see that the behavior of different acquisition functions varies according to the the function. On the Ackley function, PBGI and PBGI-D outperform most baselines, except for the non-myopic BMSEI policy in the cost-aware setting. In contrast, on the Levy function, the dynamic-λ only outperforms the EIPC baseline on small-enough cost horizons, and PBGI worse than PBGI-D the cost-aware setting: the same also holds for the BMSEI baseline, indicating that using multi-step lookahead actually reduces performance here—we will return to this

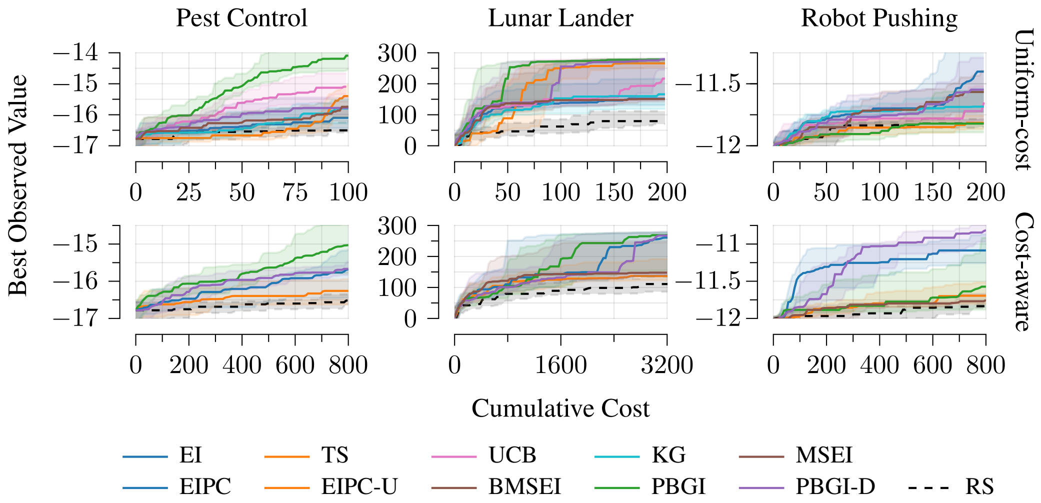

Figure 6: Empirical benchmark regret curves, shown using medians, as well as quartiles to show variability. We see in both the cost-aware and uniform-cost settings that PBGI exhibits stronger performance on the Pest Control and Lunar Lander problems, while PBGI-D together with the EI and EIPC baselines performly strongly on the Robot Pushing problem. Note that in the cost-aware variant of Robot Pushing, the non-myopic BMSEI baseline performs poorly, potentially mirroring the behavior previously seen on the unimodal Rosenbrock function in Figure 5.

momentarily in the context of the Rosenbrock function. We conclude that the constant-λ variant can in principle offer stronger performance than the dynamic-λ variant, as long as λ is not too-suboptimal, while the dynamic-λ variant is less-performant but more robust to hyperparameter choice.

We also examine performance on the unimodal, banana-shaped Rosenbrock-function. Here, both expected improvement variants perform the strongest, matching the dynamic-λ PBGI variant and multi-step lookahead baselines, and outperforming the constant-λ variant. This can intuitively be explained by the one-step optimality of expected improvement, which better-exploits the unimodal objective, while PBGI and multi-step-based acquisition functions are more conservative. We conclude that the dynamic-λ variant may be a better choice in settings where there is a potential mismatch between the objective and the prior in terms of unimodality.

4.3 Empirical objectives

Finally, we benchmark PGBI policies on three empirical global optimization problems motivated by applied challenges: *Pest Control* where $d = 25$ [23], *Lunar Lander* where $d = 12$ [8], and *Robot Pushing* where $d = 14$ [32]. Detailed descriptions of these problems and associated cost functions are in Appendix C. Note that, for Lunar Lander and Robot Pushing, the cost functions used are not automatically-differentiable: we thus apply unknown-cost PBGI and baseline variants, where the costs are modeled using a second independent Gaussian process for these objectives: the analytical form of unknown-cost PBGI variant is given in Appendix B.4.

From Figure 6, we see that the PBGI outperforms baselines on Pest Control and Lunar Lander, in both the cost-aware and uniform-cost settings. On the other hand, PBGI performs poorly on Robot Pushing, where instead expected improvement and PBGI-D perform best, and the non-myopic BMSEI baseline performs poorly. This mirrors behavior previously seen on the unimodal Rosenbrock function, from which we suspect unimodality-like behavior may be at play here as well. Note also that the performance gap between PBGI and UCB is substantially bigger here than on the Bayesian regret or synthetic problems: this may be in part because we tune UCB using the schedule of Srinivas et al. [29], which is explicitly designed for the Bayesian regret setting, and may be less-ideal for other objectives. In comparison, PBGI's tuning works reasonably well on all three problem classes simultaneously.

5 Conclusion

In this paper, we introduced a new acquisition function class for cost-aware Bayesian optimization, the *Pandora's Box Gittins index*, based on an unexplored connection between Bayesian optimization and the *Pandora's Box* problem from economics. We observed promising performance from two variants of this acquisition function class, on both cost-aware problems which are the focus of this work, and, additionally, on classical uniform-cost problems. Performance gains seen tended to be largest on higher-dimensional problems. Our work constitutes a first step toward integrating ideas from Gittins index theory, including insights from generalizations of Pandora's Box, into Bayesian optimization.

Acknowledgments

We thank James T. Wilson for his suggestions, which helped us improve the paper's presentation. AT is grateful to Anastasios Angelopoulos for hosting him at UC Berkeley on November 17th, 2022: though the visit and planned talk that day had to be cancelled last-minute due to labor strikes, the cancellation ultimately resulted in an unexpected chain of events that, months later, led to AT and ZS exchanging ideas about Gaussian processes and Gittins index theory, which ultimately led to this work. AT was supported by Cornell University, jointly via the Center for Data Science for Enterprise and Society, the College of Engineering, and the Ann S. Bowers College of Computing and Information Science. ZS was supported by the NSF under grant numbers CMMI-2307008, DMS-2023528, and DMS-2022448.

References

[1] S. Aalto, U. Ayesta, and R. Righter. On the Gittins Index in the $M/G/1$ Queue. *Queueing Systems*, 2009. Cited on page 2.

[2] M. R. Aminian, V. Manshadi, and R. Niazadeh. Markovian search with socially aware constraints. *Management Science*, 2024. Cited on pages 5, 13–15.

[3] R. Astudillo, D. Jiang, M. Balandat, E. Bakshy, and P. Frazier. Multi-step budgeted bayesian optimization with unknown evaluation costs. *Advances in Neural Information Processing Systems*, 2021. Cited on pages 1, 3, 5, 7, 12, 16–19.

[4] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 2020. Cited on pages 7, 17.

[5] S. Belakaria, J. R. Doppa, N. Fusi, and R. Sheth. Bayesian optimization over iterative learners with structured responses: A budget-aware planning approach. In *Artificial Intelligence and Statistics*, 2023. Cited on page 1.

[6] L. Doval. Whether or Not to Open Pandora's Box. *Journal of Economic Theory*, 2018. Cited on page 2.

[7] I. Dumitriu, P. Tetali, and P. Winkler. On Playing Golf with Two Balls. *SIAM Journal on Discrete Mathematics*, 2003. Cited on page 2.

[8] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek. Scalable global optimization via local Bayesian optimization. *Advances in Neural Information Processing Systems*, 2019. Cited on pages 9, 18.

[9] P. Frazier, W. Powell, and S. Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 2009. Cited on page 7.

[10] P. I. Frazier. Bayesian Optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*. 2018. Cited on page 3.

[11] R. Garnett. *Bayesian Optimization*. 2023. Cited on pages 2, 3.

[12] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 1979. Cited on page 13.

[13] J. C. Gittins, K. D. Glazebrook, and R. R. Weber. *Multi-Armed Bandit Allocation Indices*. Wiley, 2011. Cited on pages 2–4.

[14] K. D. Glazebrook and J. Niño-Mora. Parallel Scheduling of Multiclass $M/M/m$ Queues: Approximate and Heavy-Traffic Optimization of Achievable Performance. *Operations Research*, 2001. Cited on page 2.

[15] A. Gupta, H. Jiang, Z. Scully, and S. Singla. The Markovian Price of Information. In *Integer Programming and Combinatorial Optimization*, 2019. Cited on page 2.

[16] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in Neural Information Processing Systems*, 2014. Cited on page 17.

[17] S. Jiang, D. Jiang, M. Balandat, B. Karrer, J. Gardner, and R. Garnett. Efficient nonmyopic bayesian optimization via one-shot multi-step trees. *Advances in Neural Information Processing Systems*, 2020. Cited on pages 1, 7, 17.

[18] R. Kleinberg, B. Waggoner, and E. G. Weyl. Descending price optimally coordinates search. In *Economics and Computation*, 2016. Cited on page 5.

[19] E. H. Lee, D. Eriksson, V. Perrone, and M. Seeger. A nonmyopic approach to cost-constrained Bayesian optimization. In *Uncertainty in Artificial Intelligence*, 2021. Cited on page 1.

[20] Y. L. Li, T. G. Rudner, and A. G. Wilson. A study of Bayesian neural network surrogates for Bayesian optimization. In *International Conference on Learning Representations*, 2024. Cited on page 18.

[21] R. Martinez-Cantin. Bayesian optimization with adaptive kernels for robot control. In *International Conference on Robotics and Automation*, 2017. Cited on page 1.

[22] P. Milgrom and I. Segal. Envelope Theorems for Arbitrary Choice Sets. *Econometrica*, 2002. Cited on page 14.

[23] C. Oh, J. Tomczak, E. Gavves, and M. Welling. Combinatorial bayesian optimization using the graph cartesian product. *Advances in Neural Information Processing Systems*, 2019. Cited on pages 9, 18.

[24] W. Olszewski and R. Weber. A More General Pandora Rule? *Journal of Economic Theory*, 2015. Cited on page 2.

[25] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. Cited on pages 3, 19.

[26] Z. Scully, I. Grosof, and M. Harchol-Balter. The Gittins Policy is Nearly Optimal in the $M/G/k$ under Extremely General Conditions. *Measurement and Analysis of Computing Systems*, 2020. Cited on page 2.

[27] S. Singla. The Price of Information in Combinatorial Optimization. In *Symposium on Discrete Algorithms*, 2018. Cited on page 2.

[28] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 2012. Cited on pages 1, 3.

[29] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2010. Cited on pages 7, 9, 17.

[30] A. Terenin. *Gaussian Processes and Statistical Decision-making in Non-Euclidean Spaces*. PhD thesis, Imperial College London, 2022. Cited on page 12.

[31] J. v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928. Cited on page 15.

[32] Z. Wang and S. Jegelka. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning*, 2017. Cited on pages 9, 18.

[33] R. R. Weber. On the Gittins Index for Multiarmed Bandits. *The Annals of Applied Probability*, 1992. Cited on page 2.

[34] M. L. Weitzman. Optimal search for the best alternative. *Econometrica*, 1979. Cited on pages 2–5, 13, 16.

[35] X. Yue and R. A. Kontar. Why non-myopic Bayesian optimization is promising and how far should we look-ahead? A study via rollout. In *Artificial Intelligence and Statistics*, 2020. Cited on page 1.

[36] Y. Zhang, D. W. Apley, and W. Chen. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific Reports*, 2020. Cited on page 1.

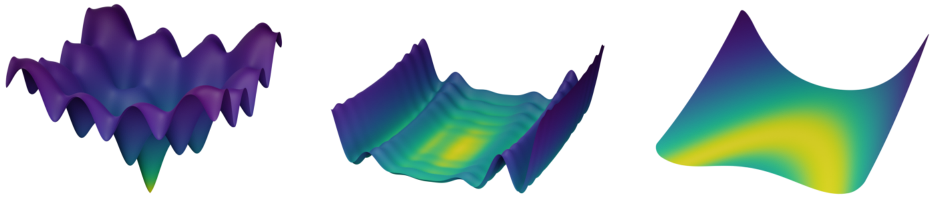
(a) Ackley (b) Levy (c) Rosenbrock

Figure 7: An illustration of the two-dimensional Ackley, Levy, and Rosenbrock functions.[1] From the visual, one can see that these functions differ in terms of multimodality and ridge-regions within the optimization landscape.

A Illustrations

Here we provide a set of additional illustrations to aid understanding of our results.

A.1 Visualization of synthetic benchmark functions

To better understand how the behavior of Bayesian optimization algorithms on the three different synthetic benchmark functions might be affected by their geometric shape, Figure 7 provides a visual illustration of their two-dimensional variants.[1] This allows us to visually see the multimodality of the Ackley function, multimodality and ridge-like regions in the Levy function, and unimodality of the Rosenbrock function. While we use higher-dimensional versions of these in our experiments, this illustration provides some intuition for what the resulting the optimization landscape might look like, helping contextualize results.

A.2 EI/EIPC-U performance counterexample

In Section 3.3, we showed a Bayesian optimization problem on which expected improvement per unit cost (EIPC) has poor performance. Here, we show that this problem can be modified so that ordinary expected improvement (EI), which ignores the cost function, also has poor performance—a somewhat obvious, but nonetheless important sanity check that we make to ensure that costs play a sufficiently-important role in problems of this class to merit their consideration. This is shown in Figure 8. It is not hard to construct a less-visualization-friendly variant of these problems on which both expected improvement per unit cost and ordinary expected improvement perform poorly, by considering cost functions which are appropriately-weighted sums of bump functions.

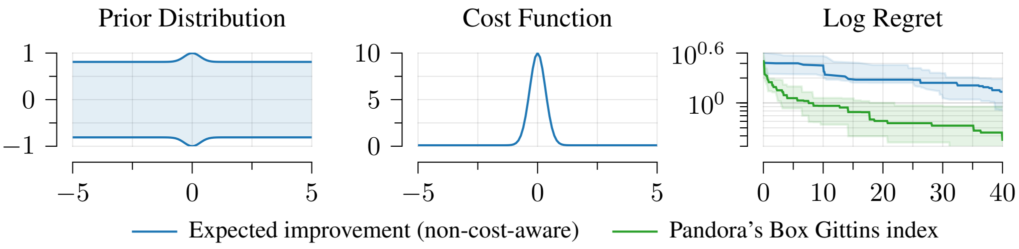

Figure 8: A Bayesian optimization problem with heterogeneous costs on which expected improvement, which ignores the cost function, has poor performance. Like the EIPC example of Figure 2, the construction also mirrors Astudillo et al. [3], Section A. The domain is $X = [-500, 500]$, which we visualize on the subinterval $[-5, 5]$. Left: illustration of the non-uniform prior variance, which is given by a Matérn-5/2 kernel scaled by a narrow bump function. Center: the cost function, which is a narrow bump-shaped function. Right: regret curves.

[1] This visual originally appeared in Terenin [30], and is reproduced here with permission.

B Theory and Calculations

Below, we provide additional ideas to help understand the Pandora's Box problem and acquisition function that results from its considerations.

B.1 Additional intuition on Pandora's Box

In what follows, we sketch a viewpoint from which one can see the key idea behind why Proposition 1 holds. Rather than considering the full Pandora's Box problem with a general set of open and closed boxes, consider first the case where there is exactly one closed and one open box. To slightly simplify notation, let f denote the random reward inside the closed box, let c denote the cost of opening the closed box, assumed deterministic, and let g denote the visible reward of the open box. Our possible actions are as follows:

1. *Open the closed box.* In this case, we pay a cost of c, but subsequently get to choose between taking the realized value f, or instead taking g from the box that was originally open. In expectation, the total value obtained by taking this action is $\mathbb{E}(\max(f, g)) - c$.
2. Take the reward from the open box. The total value obtained is g.

We can therefore analytically solve for the optimal policy of this respective Markov decision process: we open the closed box if $\mathbb{E}(\max(f, g)) - c \geq g$, and take the reward from the open box if $\mathbb{E}(\max(f, g)) - c \leq g$, with both actions optimal in the case of equality. As consequence, if g is such that both actions are optimal, the same value is obtained no matter whether one chooses to open the box or not. Rewriting the preceding expressions slightly, this occurs when

$$\mathbb{E}\max(f - g, 0) = c \tag{7}$$

where in the case of multiple boxes the left-hand-side becomes the expected improvement function. The insight of Weitzman [34]—and indeed of Gittins [12] in a much more general setting—is that one can modify the Pandora's Box Markov decision process by replacing closed boxes with open boxes whose value g satisfies (7) without changing the optimal policy. As a consequence, the optimal policy is precisely the Gittins index policy of Proposition 1.

As a final point, note that the assumption that c is deterministic is made without loss of generality: if c is instead stochastic but has finite expectation, the same reasoning applies, but with the value c in (7) replaced with its expected value. We will make use of this in Appendix B.4.

B.2 Relationship between expected budget-constrained and cost-per-sample problems

For completeness, since Aminian et al. [2], Theorem 1 is stated in somewhat-different terminology than our work and includes additional technical restrictions which simultaneously (a) rule out Gaussian-distributed rewards, and, at the same time, (b) are not used for proving the claim we are interested in but instead applied elsewhere, we now prove Proposition 2.

In what follows, recall that the assumptions of Proposition 1 are (i) X is discrete, (ii) $\mathbb{E}\,|f(x)| < \infty$ for all x, (iii) $f(x)$ and $f(x')$ are independent for $x \neq x'$. Further, Proposition 1 was stated for deterministic costs: more generally, we allow for stochastic costs satisfying $0 < \mathbb{E}(c(x)) < \infty$, and in such cases replace the deterministic costs in the definition of $\alpha_\star$ with the expected costs.

Proposition 2 (Corollary of Aminian et al. [2], Theorem 1)**.** *Consider the expected budget-constrained problem, with the assumptions of Proposition 1. Assume $B > \inf_{x \in X} c(x)$. Then there exists a $\lambda > 0$ such that the policy defined by maximizing the Gittins index acquisition function $\alpha_\star(\cdot)$ with its associated stopping rule, defined using costs $\lambda c(x)$, is Bayesian-optimal.*

Proof. For clarity, we present the proof initially for the uniform-cost case where the cost function $c(x) \equiv 1$. The argument for the cost-aware case can be extended similarly with modifications.

We begin by formalizing this proposition. A deterministic policy with stopping rule $\pi : \mathcal{H} \to \mathcal{X} \cup \{\Delta\}$ maps the current history to either a point in the domain or a decision to stop (denoted by Δ). Here $\mathcal{H} = \bigcup_{t=0}^{\infty} (\mathbb{R}^d \times \mathbb{R})^t$ denotes the space of all possible histories. Let $\Pi_D := \{\pi : \pi(\varnothing) \neq \Delta\}$ be the set of all possible deterministic policies that evaluate at least one point. Given a deterministic policy $\pi \in \Pi_D$, the histories and decisions update recursively as follows:

- $\pi(H_{t-1}) \neq \Delta$: $x_t = \pi(H_{t-1})$, $H_t = H_{t-1} \cup \{(x_t, f(x_t)\}$
- $\pi(H_{t-1}) = \Delta$: $H_t = H_{t-1}$

The stopping time of a policy π is a random variable given by $\tau(\pi) = \inf\{t > 0 : \pi(H_t) = \Delta\}$. For simplicity, we will occasionally write τ without (π). The evaluation time budget is specified by T. Let Π be the set of all possible randomized policies, defined as the probability measures on Π_D. The expected best observed value under a policy $\pi \in \Pi$ is given by $V^\pi := \mathbb{E}^\pi[\sup_{t\leq\tau} f(x_t)]$. We then define the feasible set of policies and the optimal value achieved for the expected budget-constrained and the cost-per-sample optimization as follows:

- *Expected budget-constrained optimization*:

$$\Pi_{\text{ebc}} := \{\pi \in \Pi : \mathbb{E}^\pi[\tau] \leq T\}, \quad V_{\text{ebc}} := \sup_{\pi\in\Pi_{\text{ebc}}} V^\pi.$$

- *Cost-per-sample optimization*:

$$V_{\text{cps}}(\lambda) = \sup_{\pi\in\Pi}\{V^\pi - \lambda\mathbb{E}^\pi[\tau]\}.$$

Finally, Proposition 2 establishes that there exists a $\lambda > 0$ such that the optimal policy for the cost-per-sample optimization with this λ, denoted $\pi^*(\lambda)$, is also Bayesian-optimal for the expected budget-constrained optimization.

We show the existence of such λ by construction. In the following, we will demonstrate that the infimum $\lambda^* \in \operatorname{arginf}_{\lambda>0}\{V_{\text{cps}}(\lambda)+\lambda T\}$ satisfies the condition of Proposition 2. Specifically, we will show that 1) $\pi^*(\lambda^*)$ meets the expected budget constraint, meaning $\pi^*(\lambda^*) \in \Pi_{\text{ebc}}$, and 2) $\pi^*(\lambda^*)$ attains the optimal value of the expected budget-constrained optimization, i.e., $V^{\pi^*(\lambda^*)} = V_{\text{ebc}}$.

To establish 1), we demonstrate that: i) an infimum exists for $V_{\text{cps}}(\lambda) + \lambda T$, and ii) the expected stopping time under the policy $\pi^*(\lambda^*)$ equals T, i.e., $\mathbb{E}^{\pi^*(\lambda^*)}[\tau] = T$, and therefore $\pi^*(\lambda^*) \in \Pi_{\text{ebc}}$.

For i), observe that the function $V_{\text{cps}}(\lambda) + \lambda T$ is convex in λ. This follows as it is expressed as $\sup_{\pi\in\Pi}\{V^\pi + \lambda(T - \mathbb{E}^\pi[\tau])\}$, which is a supremum over a collection of affine functions of λ. Additionally, it can be bounded from below by an affine function of λ with a non-negative slope since $V_{\text{cps}}(\lambda) + \lambda T \geq V^{\pi_1} + \lambda(T - \mathbb{E}^{\pi_1}[\tau]) \geq V^{\pi_1} + \lambda(T - 1) = \mathbb{E}[f(x_1)] + \lambda(T - 1)$ where π_1 is a policy that always selects the same point x_1 at the beginning and stops immediately at $\tau = 1$. The convexity and boundedness guarantee the existence of a finite infimum λ^*.

For ii), let $N = |X|$ be the number of points in the finite set X. Since every policy evaluates at least one point and at most N points, it follows that $1 \leq \mathbb{E}^{\pi^*(\lambda^*)}[\tau] \leq N$. According to the *Envelope Theorem for arbitrary choice sets* [22, Theorem 1], if the value function $V_{\text{cps}}(\lambda)$ is differentiable with respect to the parameter λ, then its derivative is given by:

$$V'_{\text{cps}}(\lambda) = \frac{\partial(V^{\pi^*(\lambda)} - \lambda\mathbb{E}^{\pi^*(\lambda)}[\tau])}{\partial\lambda} = -\mathbb{E}^{\pi^*(\lambda)}[\tau].$$

Since the function $V_{\text{cps}}(\lambda) + \lambda T$ is convex in λ and its derivative $T - \mathbb{E}^{\pi^*(\lambda)}[\tau]$ increases from $T - N$ to $T - 1$ as λ grows, then its infimum λ^* is achieved when the derivative equals to 0, i.e., $T - \mathbb{E}^{\pi^*(\lambda^*)}[\tau] = 0$. Note that here we assume differentiability in λ^* as randomization over tie-breaking rules can be employed to restore differentiability when $V_{\text{cps}}(\lambda)$, optimized over the deterministic policies π_D, is non-differentiable. For further details, see Aminian et al. [2].

We now proceed to establish 2). The definition of $\pi^*(\cdot)$ implies that $V^{\pi^*(\lambda^*)} - \lambda^*\mathbb{E}^{\pi^*(\lambda^*)}[\tau] = \sup_{\pi\in\Pi}\{V^\pi - \lambda^*\mathbb{E}^\pi[\tau]\} = V_{\text{cps}}(\lambda^*)$. Given our earlier result that $\mathbb{E}^{\pi^*(\lambda^*)} = T$, it follows that $V^{\pi^*(\lambda^*)} = V_{\text{cps}}(\lambda^*) + \lambda^* T$. By the definition of λ^*, this is exactly $\inf_{\lambda>0}\{V_{\text{cps}}(\lambda) + \lambda T\}$, implying $V^{\pi^*(\lambda^*)} = \inf_{\lambda>0}\{V_{\text{cps}}(\lambda) + \lambda T\}$. Since we have also demonstrated that $\pi^*(\lambda^*)$ satisfies the expected budget constraint, and hence $V^{\pi^*(\lambda^*)} \leq V_{\text{ebc}}$, it remains to show that

$$V_{\text{ebc}} \leq \inf_{\lambda>0}\{V_{\text{cps}}(\lambda) + \lambda T\}. \tag{8}$$

Last, we prove Eq. (8). The expected budget-constrained optimization can be transformed into an unconstrained optimization using Lagrangian relaxation, then

$$\begin{aligned} V_{\mathrm{ebc}} &= \sup_{\pi \in \Pi_{\mathrm{ebc}}} V^{\pi} \\ &= \sup_{\pi \in \Pi} \inf_{\lambda > 0} \{V^{\pi} + \lambda(T - \mathbb{E}^{\pi}[\tau])\} \\ &\leq \inf_{\lambda > 0} \sup_{\pi \in \Pi} \{V^{\pi} + \lambda(T - \mathbb{E}^{\pi}[\tau])\} \\ &= \inf_{\lambda > 0} \{V_{\mathrm{cps}}(\lambda) + \lambda T\}, \end{aligned}$$

where the inequality arises from the *max-min inequality* [31]. It can be validated by the following standard argument: $\forall\, \pi \in \Pi, \lambda > 0$, it holds that

$$\inf_{\lambda' > 0} \{V^{\pi} + \lambda'(T - \mathbb{E}^{\pi}[\tau])\} \leq V^{\pi} + \lambda(T - \mathbb{E}^{\pi}[\tau]) \leq \sup_{\pi' \in \Pi} \{V^{\pi'} + \lambda(T - \mathbb{E}^{\pi'}[\tau])\}. \tag{9}$$

Taking the supremum over all policies $\pi \in \Pi$ on the left and the infimum over all $\lambda > 0$ on the right confirms the max-min inequality, completing the proof. □

Remark 1: To adapt the preceding proof to the cost-aware case, the following modifications can be made: let B be the evaluation cost budget, now the expected budget constraint should be $\mathbb{E}^{\pi}\left[\sum_{t \leq \tau} c(x_t)\right] \leq B$. We can then replace all τ with $\sum_{t \leq \tau} c(x_t)$ and all T with B in the proof, and adjust the range of $\mathbb{E}^{\pi^*(\lambda^*)}[\tau]$ to be $[\min_{x \in X} c(x), \sum_{x \in X} c(x)]$ instead of $[1, N]$.

Remark 2: The objective of our expected budget-constrained problem differs from that presented in Aminian et al. [2]: while we focus on minimizing the simple regret, their work aims at maximizing the utility. Their definition of utility is more general and one particular relevant example is the best observed value minus the cumulative costs.

Remark 3: We can also define the feasible policy set and the optimal value of the almost-sure budget-constrained problem:

$$\Pi_{\mathrm{asbc}} := \{\pi \in \Pi : \mathbb{P}^{\pi}(\tau \leq T) = 1\}, \quad V_{\mathrm{asbc}} := \sup_{\pi \in \Pi_{\mathrm{asbc}}} V^{\pi}.$$

Then this optimal value is upper-bounded by the optimal value of the expected budget-constrained optimization: $V_{\mathrm{asbc}} \leq V_{\mathrm{ebc}}$ since $\Pi_{\mathrm{asbc}} \subseteq \Pi_{\mathrm{ebc}}$.

B.3 Gradient of the PBGI acquisition function

Gradient-based methods including multi-start stochastic gradient descent, BFGS, and L-BFGS-B effectively optimize analytical acquisition functions, such as EI and UCB. These methods are also applicable for optimizing the PBGI acquisition function. To facilitate this, we provide the gradient formula for the PBGI acquisition function. In what follows, mirroring the preceding sections, if costs are stochastic then $c(x)$ should be replaced with its respective mean.

Proposition 3 (Gradient of PBGI)**.** *Let $\mu(x)$ and $\sigma(x)$ be the mean and standard deviation of the posterior Gaussian process $(f \mid y_1, .., y_t)(x)$. With this notation, the gradient of the acquisition function $\alpha_{\mathrm{PBGI}}(x)$ is given by*

$$\nabla \alpha_{\mathrm{PBGI}}(x) = \nabla \mu(x) + \frac{\phi\left(\frac{\mu(x) - \alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\right) \nabla \sigma(x) - \lambda \nabla c(x)}{\Phi\left(\frac{\mu(x) - \alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\right)}, \tag{10}$$

where ϕ and Φ denote the density and cumulative distribution function of a standard normal distribution, respectively.

Proof. Recall that when $\psi(x) \sim \mathcal{N}(\mu(x), \sigma(x))$ is a Gaussian distribution, the expected improvement with respect to the comparator y is given as

$$\mathrm{EI}_{\psi}(x; y) = (\mu(x) - y)\Phi\left(\frac{\mu(x) - y}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\mu(x) - y}{\sigma(x)}\right). \tag{11}$$

Next, note by definition of α_{PBGI}, we have

$$\mathrm{EI}_{f|y_1,..,y_t}(x;\alpha_{\mathrm{PBGI}}(x)) = \lambda c(x). \tag{12}$$

Differentiating this with respect to x on both sides gives

$$\nabla\,\mathrm{EI}_{f|y_1,..,y_t}(x;\alpha_{\mathrm{PBGI}}(x)) = \lambda\nabla c(x). \tag{13}$$

Applying the product and chain rule to the left-hand-side gives

$$\nabla\,\mathrm{EI}_{f|y_1,..,y_t}(x;\alpha_{\mathrm{PBGI}}(x)) = (\nabla\mu(x) - \nabla\alpha_{\mathrm{PBGI}}(x))\Phi\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg) \tag{14}$$

$$+ (\mu(x)-\alpha_{\mathrm{PBGI}}(x))\phi\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg)\nabla\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg) \tag{15}$$

$$+ \nabla\sigma(x)\phi\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg) \tag{16}$$

$$+ \sigma(x)\phi'\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg)\nabla\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg). \tag{17}$$

Recall the identity for the derivative of the Gaussian density, namely

$$\phi'(x) = -x\phi(x). \tag{18}$$

Applying this identity to (17) gives

$$\sigma(x)\phi'\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg)\nabla\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg) \tag{19}$$

$$= -(\mu(x)-\alpha_{\mathrm{PBGI}}(x))\phi\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg)\nabla\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg) \tag{20}$$

which is equal to the negation of (15), hence (15) and (17) cancel: we get

$$(\nabla\mu(x)-\nabla\alpha_{\mathrm{PBGI}}(x))\Phi\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg)+\nabla\sigma(x)\phi\bigg(\frac{\mu(x)-\alpha_{\mathrm{PBGI}}(x)}{\sigma(x)}\bigg) = \lambda\nabla c(x). \tag{21}$$

Rearranging this gives the expression in the claim. □

B.4 Closed-form expression for the PBGI unknown-cost variant

In the Lunar Lander and Robot Pushing empirical examples of Section 4, the cost function does not admit an analytic, automatically-differentiable form, and can only be evaluated in a black-box manner. To handle this, we model the logarithm of the costs as a Gaussian process, and condition this process on the costs observed at locations evaluated so far. This mirrors how unknown costs are handled in other acquisition functions, such as the budgeted multi-step expected improvement acquisition function of Astudillo et al. [3].

From the viewpoint of the Pandora's Box problem, stochastic costs make little difference: following the discussion in Appendix B.1, the optimality results of Weitzman [34] continue to hold even if costs are stochastic, so long as the costs in the formula for $\alpha_\star$ are replaced with expected costs. Mirroring this, if we plug in the mean of a log-normal random variable into the definition of α_{PBGI}, we obtain the following acquisition function.

Definition 4. *Let $c(x)$ be log-normal for all x. For a dataset $(x_1, y_1), .., (x_t, y_t)$, let $\mu_{\ln c}$ and $\sigma_{\ln c}$ be the posterior mean and posterior standard deviation of the log-costs. Define the* UNKNOWN-COST PANDORA'S BOX GITTINS INDEX ACQUISITION FUNCTION *by*

$$\alpha_{\text{PBGI-U}}(x) = g \qquad \textit{where } g \textit{ solves} \qquad \mathrm{EI}_{f|y_1,..,y_t}(x;g) = \lambda\exp\bigg(\mu_{\ln c}(x) + \frac{\sigma_{\ln c}(x)^2}{2}\bigg). \tag{22}$$

The interpretation of λ, namely as a hyperparameter that determines the expected budget the algorithm will use before reaching its respective stopping time, is the same as in the known-cost setting. One can define cost-per-sample and anytime variants in the same manner as well.

C Experimental Setup

We implement all experiments in BoTorch. Following standard practice, we initialize each optimization algorithm with $2(d+1)$ values drawn using a quasirandom Sobol sequence, where d is the dimension of the domain. All computations were run on CPU, with individual experiments ran in parallel on various nodes of the Cornell G2 cluster, each allocated up to 4GB of memory. Exceptions include KG, MSEI, and BMSEI for higher dimensions, which required substantially more memory, up to 32GB. Most individual runs took several minutes at most, with exception of the more-expensive KG, MSEI and BMSEI baselines: more information with a direct runtime comparison is given in Appendix D.

Gaussian process models. For the Gaussian process prior, we use Matérn kernels with fixed hyperparameters, namely smoothness $5/2$ and length scale 10^{-1}: additional experimental results which show the effect of varying these choices are given in Appendix D. For the synthetic and empirical experiments, we standardize our data to be zero mean and unit variance, following BoTorch defaults. To maintain consistency, we do not standardize data in the Bayesian regret experiments. In the unknown-cost experiments, we model the objective and the logarithm of the cost function using independent Gaussian processes.

Acquisition function optimization. This is done as follows. We begin by computing acquisition values at $200d$ points spread across the domain X, where d is the dimension of X. For all acquisition functions except MSEI and BMSEI, which use a modification described below, the initial $200d$ points are generated using a Sobol sequence design. From these, $10d$ points are selected according to the initialization heuristic used by BoTorch, detailed in Balandat et al. [4], Appendix F.1. We then use multi-start L-BFGS-B to optimize the acquisition function from each selected point. The point with highest acquisition value among the $10d$ optimized points is chosen as the next evaluation point.

We now detail the modified strategy used for MSEI and BMSEI: here, the initial $200d$ points are selected using the warm-start initialization strategy described in Jiang et al. [17], Appendix D and Astudillo et al. [3], Appendix F. This strategy uses the optimal solution from the previous iteration, targeting the branch that originates from the tree's root and whose fantasy sample most closely matches the actual observed value of the previously suggested candidate on the true function. This modification favors MSEI and BMSEI, slightly disadvantaging PBGI and other baselines.

Acquisition function hyperparameters. For PBGI, we use $\lambda = 10^{-4}$ and a total of 100 iterations of bisection search without any early stopping or other performance and reliability optimizations. For UCB, we follow the schedule in Srinivas et al. [29], Theorem 1 given by $\beta_t = 2\log(dt^2\pi^2/6\delta)$, where d is the number of dimensions. We also adopt the choice of $\delta = 0.1$ and a scale-down factor of 5, as used in that work's experiments. For MSEI and BMSEI, we use 4 lookahead steps, each with a batch size of 1 and a single fantasy point.

Omitted baselines. We omit MSEI from the Bayesian regret plots for $d = 16$ with $\kappa = 10^{-1}$ and for $d = 32$ with all length scale choices because we were unable to get it to work reliably in these settings: the implementation of Jiang et al. [17] results in frequent crashes due to running out of memory and related issues when used on higher-difficulty problems. We also omit KG from $d = 32$ and BMSEI from the cost-aware Pest Control and Robot Pushing experiments for the same reasons.

In addition to the baselines mentioned in Section 4, we also implemented the *predictive entropy search (PES)* acquisition function of Hernández-Lobato et al. [16], but could not get its computations to run reliably in an automatic-differentiation-based environment without resulting in NaN gradients. Hernández-Lobato et al. [16] document this behavior, and suggest using finite-differencing in situations where it occurs: however, from initial examination, we found this to decrease performance on higher-dimensional problems. We therefore opted to restrict ourselves to automatically-differentiable baselines and omit PES, to ensure that performance differences seen can reliably be attributed to the acquisition functions used, and not to gradient computation.

Objective functions: Bayesian regret. For Bayesian regret, this is straightforward: the objective is simply a draw from a Fourier feature approximation of the respective Gaussian process prior, drawn in such a way that different baselines with the same random number seed share the same objective, but objectives for different seeds are different draws from the same prior. We use a total of 1024 Fourier features.

Objective functions: synthetic benchmarks. The synthetic benchmarks we use are as follows.

Ackley: this is

$$f_{\mathrm{A}}(x_1,..,x_d) = 20 - 20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + e \tag{23}$$

with search domain $X = [-1,1]^d$.

Levy: this is

$$f_{\mathrm{L}}(x_1,..,x_d) = s_1 + \sum_{i=1}^{d-1}(w_i-1)^2\big(1+10\sin(\pi w_i+1)^2\big) + (w_d-1)^2\big(1+\sin(2\pi w_d)^2\big) \tag{24}$$

with $w_i = 1 + \frac{x_i - 1}{4}$ and $s_1 = \sin(\pi w_1)^2$, and search domain $X = [-10,10]^d$.

Rosenbrock: this is

$$f_{\mathrm{R}}(x_1,..,x_d) = \sum_{i=1}^{d-1}\big(100(x_{i+1}-x_i^2)^2 + (x_i-1)^2\big) \tag{25}$$

with search domain $X = [-5,10]^d$. Specifically, we tested dimension $d = 4, 8, 16$ in our synthetic benchmark experiments. In the cost-aware Bayesian regret and synthetic benchmark experiments, we use the cost function

$$c(x) = 20\|S(x)\|_1 + 1 \tag{26}$$

where S is an affine map used to standardize the input domain: specifically, $S(x) = Ax + b$ where A is a diagonal matrix and b is a vector, both chosen so that the image of X under S is $[0,1]^d$.

Objective functions: empirical. We now detail the empirical objective functions.

Pest Control ($d = 25$). The pest control problem, as described in Oh et al. [23] aims to minimize the spread of pests as well as the costs of prevention treatment. We adopt the experiment setup from Li et al. [20], framing this as a categorical optimization problem with 25 variables, each representing a stage of intervention with 5 values reflecting different treatments. The objective function combines the spread of pests and the costs of prevention. In our cost-aware experiment, we use the cost of prevention as the cost function. This can be computed in an automatically-differentiable manner, thus this problem is a known-cost problem.

Lunar Lander ($d = 12$). Following the setup in Eriksson et al. [8], we consider a reinforcement learning problem optimizing a controller for the lunar lander as implemented in OpenAI Gym, which includes 12 continuous input dimensions for engine throttle adjustments. The state space captures the lander's position, angle, time derivatives, and leg contact status. The controller's actions allow for directional booster firings or inaction. The objective is to maximize the average final reward over 50 randomly generated environments. For cost-aware experimentation, we choose the cost to be the average number of simulation time steps, assuming batch processing in groups of 16. This assumption is based on the implementation found in the code associated with Li et al. [20]. Note that this objective involves the number of actual simulation steps used, and is therefore not automatically-differentiable: and we thus consider this problem to be an unknown-cost problem.

Robot Pushing ($d = 14$). In this work, we adapt one of the three versions of the robot pushing problem designed by Wang and Jegelka [32], where two robots work to push two objects to their specified targets. The problem's complexity is captured through 14 control parameters, including each robot's initial placement and motion settings. The objective minimizes the sum of the distances from the final position of each object to its respective target. In our cost-aware experiments, we test both known-cost and unknown-cost variants. The known-cost variant is the maximum of the two robots' operational duration and the unknown-cost variants variant is the sum of their traversal distances representing the total energy use, similar to the cost function used for energy-aware robot pushing benchmark of Astudillo et al. [3], but with one modification: we use the distance traversed by the robot arms instead of the distance the objects being moved. To understand the effect of this difference, we include the results for both the unknown-cost and known-cost versions in Figure 10.

D Additional Experimental Results

Here, we provide additional experimental results to better understand performance differences and other aspects of policy behavior, including the effect of various problem hyperparmeters.

D.1 Runtime comparison

Here, we provide a runtime comparison between PBGI and various baselines, including inexpensive baselines such as EI and TS, and expensive ones such as MSEI. We do so in the the Ackley synthetic benchmark setting, using the same hyperparameter settings as the main experiments. We measure the time to compute and optimize the acquisition function.

Results can be seen in Figure 9. We see that PBGI is very slightly slower than EI and TS, but significantly faster than either KG or MSEI, though the runtime of the latter decreases substantially as it accumulates more data. Overall, we conclude that PBGI's runtime is closer to that of classical acquisition functions than sophisticated lookahead-based variants.

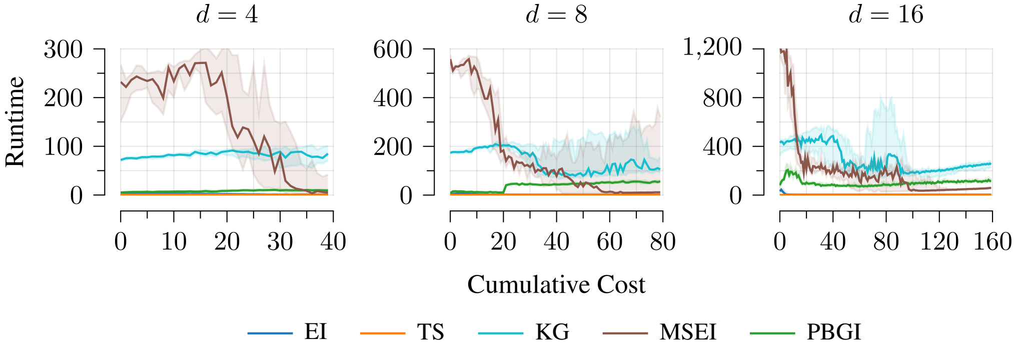

Figure 9: Runtime comparison of PBGI against baselines for computing the acquisition function on the Ackley benchmark across different dimensions ($d = 4, 8, 16$). We see that runtime of PBGI is slightly slower than EI and TS, but significantly faster than KG and MSEI.

D.2 Effect of unknown costs in Robot Pushing

The Robot Pushing empirical benchmark involves two cost functions: a known-cost variant representing total operational duration, and an known-cost variant representing total distance traversed, a proxy for energy use similar to the variant considered by Astudillo et al. [3]. One can therefore ask: how different is the resulting algorithm behavior in these two settings? Figure 10 shows this: it reveals that for the known-cost variant, EIPC and PBGI-D perform similarly, whereas for the unknown-cost variant, PBGI-D achieves the best performance on all except the shortest time horizons, where EIPC is instead competitive. Other baselines, most notably BMSEI, substantially underperform EIPC and PBGI-D, behaving similarly in both settings.

D.3 Kernel and problem hyperparameters

Choice of kernel. To check whether our results are sensitive to the kernel used for the Gaussian process model, we replicated the Bayesian regret experiments with Matérn kernels with smoothness parameters $\nu = 3/2, 5/2$, as well as the squared exponential kernel, which is the limit of Matérn kernels as $\nu \to \infty$ [25].

Similar to the original results of Figure 4, we can clearly see from Figure 11 and Figure 12 that behavior splits into three regimes:

1. *Easy:* d sufficiently-small, most policies achieve similar performance.
2. *Medium-hard:* d moderate-to-large, both PBGI variants perform better than baselines.
3. *Very hard:* d sufficiently large, no policy outperforms random search.

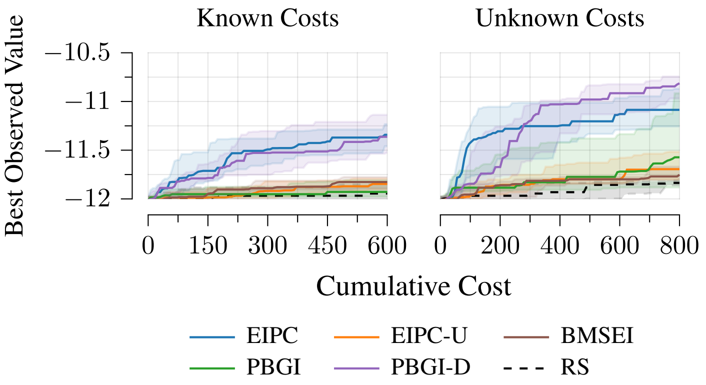

Figure 10: Experimental results for the Robot Pushing empirical benchmark, with the known-cost variant (left) and unknown-cost variant (right). We see that performance overall is similar, with EIPC and PBGI-D performing strongest. Their relative performance is similar in the known-cost variant, whereas in the unknown-cost variant PBGI-D outperforms EIPC on sufficiently-large horizons, and vice-versa on sufficiently small horizons.

We also see that $d = 32$ lands in the very-hard regime for the uniform-cost case but not for the cost-aware case: intuitively, this occurs because costs can reduce the effective volume of the search space, since high-cost regions without promising points can be excluded from search.

This behavior is consistent among different kernels, but where the exact threshold at which regimes switch differs. In particular, for the squared exponential kernel, the separation between the medium and the hard regime appears earlier than for the other variants: all policies there have similar performance to random search when $d = 16$.

Choice of length scale. To check whether our results are sensitive to the Gaussian process model's length scale, we compare $\kappa = 10^{-1}$ with $\kappa = 5 \times 10^{0}$ and $\kappa = 10^{0}$. From Figure 13 and Figure 14, which show uniform-cost and cost-aware results, respectively, we can see that PBGI variants also have much better performances as the dimension increases in both scenarios. Since $\kappa = 10^{0}$ and $\kappa = 5 \times 10^{-1}$ result in easier problems than $\kappa = 10^{-1}$, in the uniform-cost case $d = 32$ lands into the medium-hard regime rather than the very-hard regime.

Synthetic benchmark dimension. To better understand the effect of problem dimension in settings outside of Bayesian regret, we repeat the synthetic benchmark experiments with $d = 4$, $d = 8$ and $d = 16$. Results in Figure 16. Since $d = 4$ and $d = 8$ are easier to solve, here PBGI variants perform comparably to baselines.

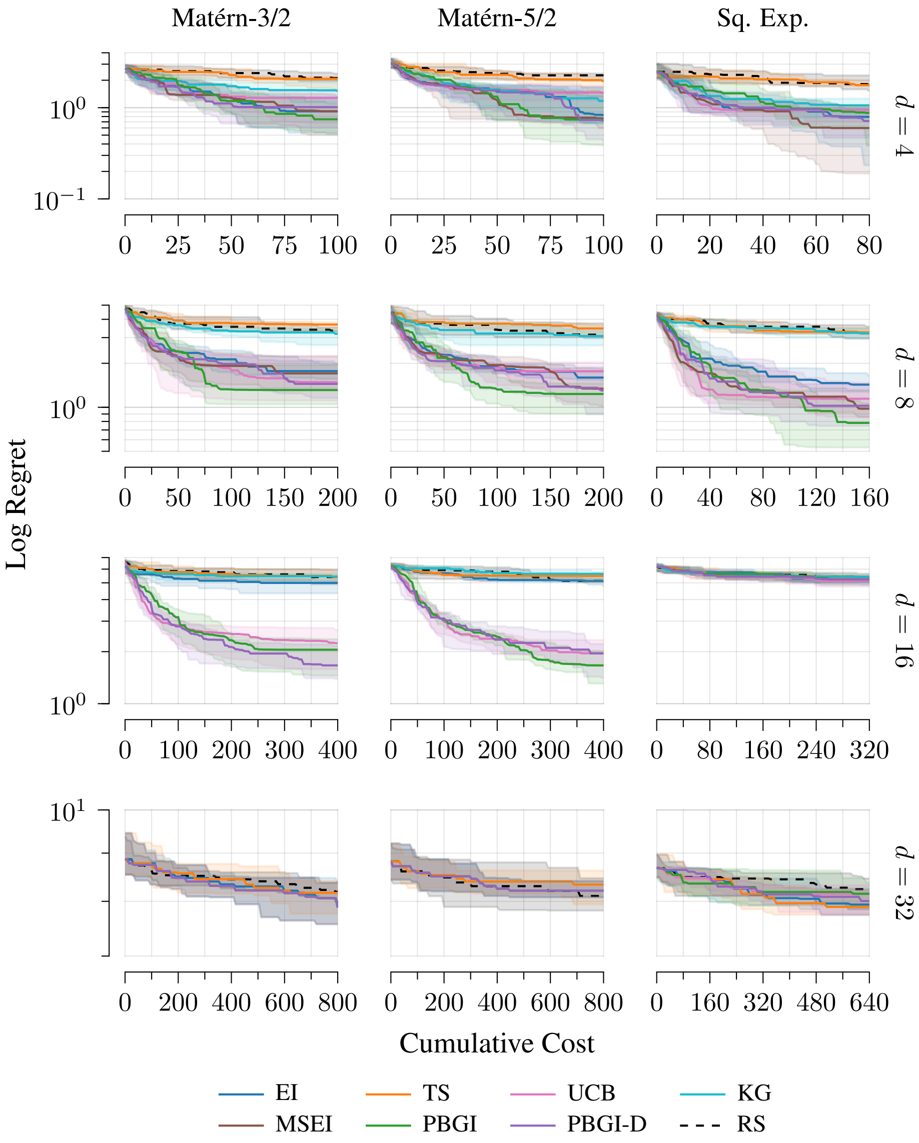

Figure 11: Comparison of Bayesian regret across Gaussian process priors with different kernels over different dimensions, in the uniform-cost setting. All length scales are $\kappa = 10^{-1}$. We see that overall behavior is similar, but the precise thresholds at which each example switches between the easy, medium-hard, and very hard regimes differ.

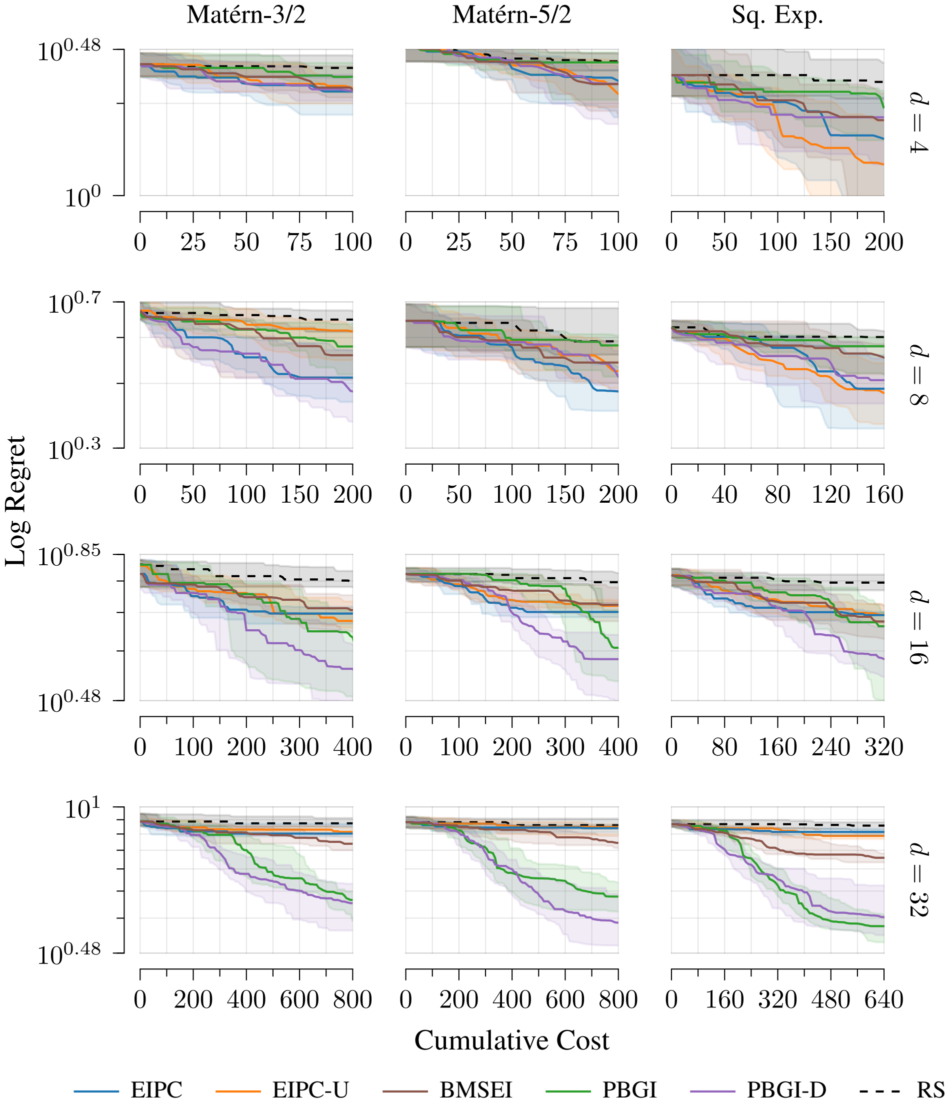

Figure 12: Comparison of Bayesian regret across Gaussian process priors with different kernels over different dimensions, in the cost-aware setting. All length scales are $\kappa = 10^{-1}$. We see that overall behavior is similar, but the precise thresholds at which each example switches between the easy, medium-hard, and very hard regimes differ.

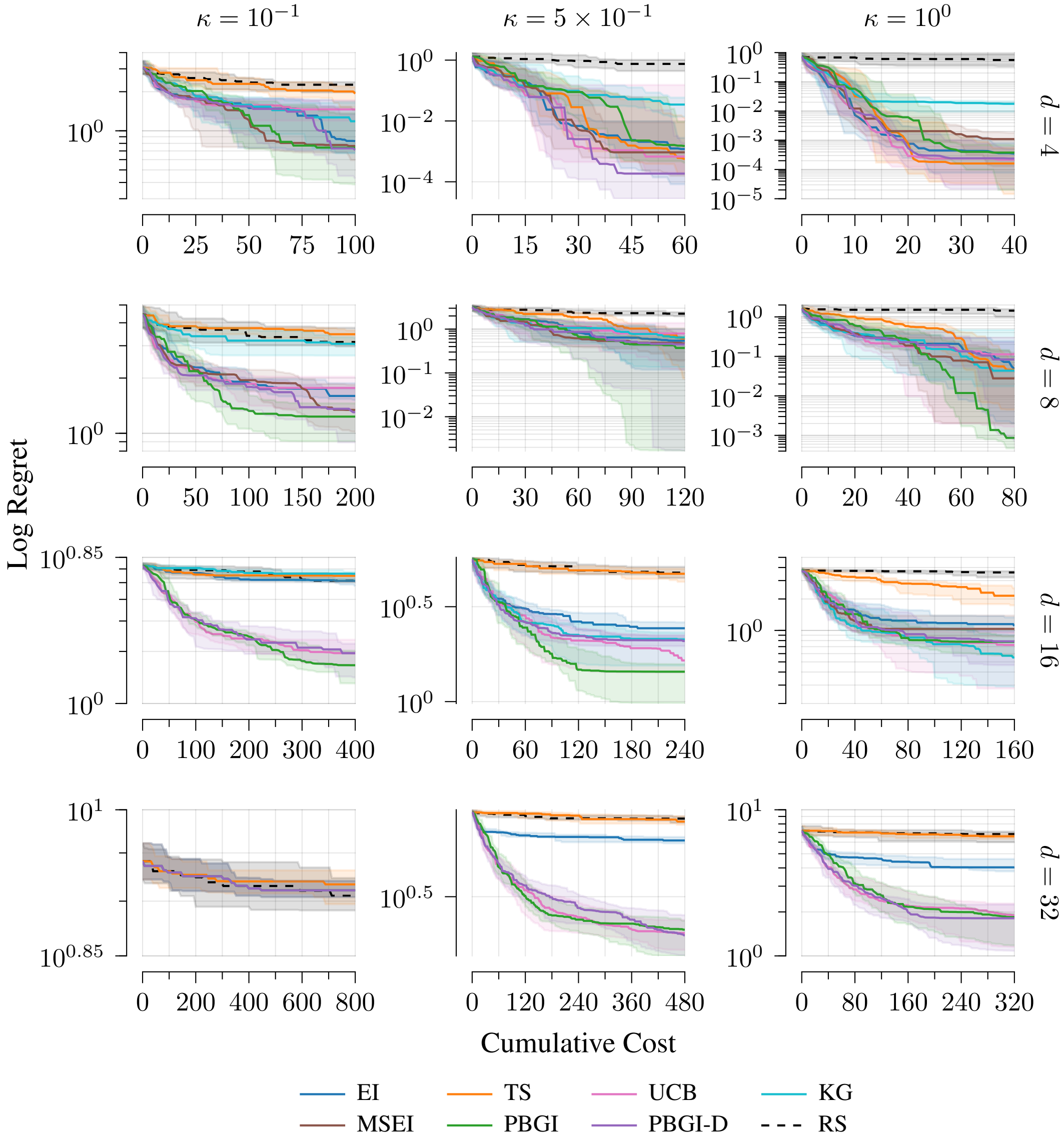

Figure 13: Comparison of Bayesian regret across different length scales and dimensions, with a Matérn-5/2 kernel, in the uniform-cost setting. We see that overall behavior is similar, but the precise thresholds at which each example switches between the easy, medium-hard, and very hard regimes differ.

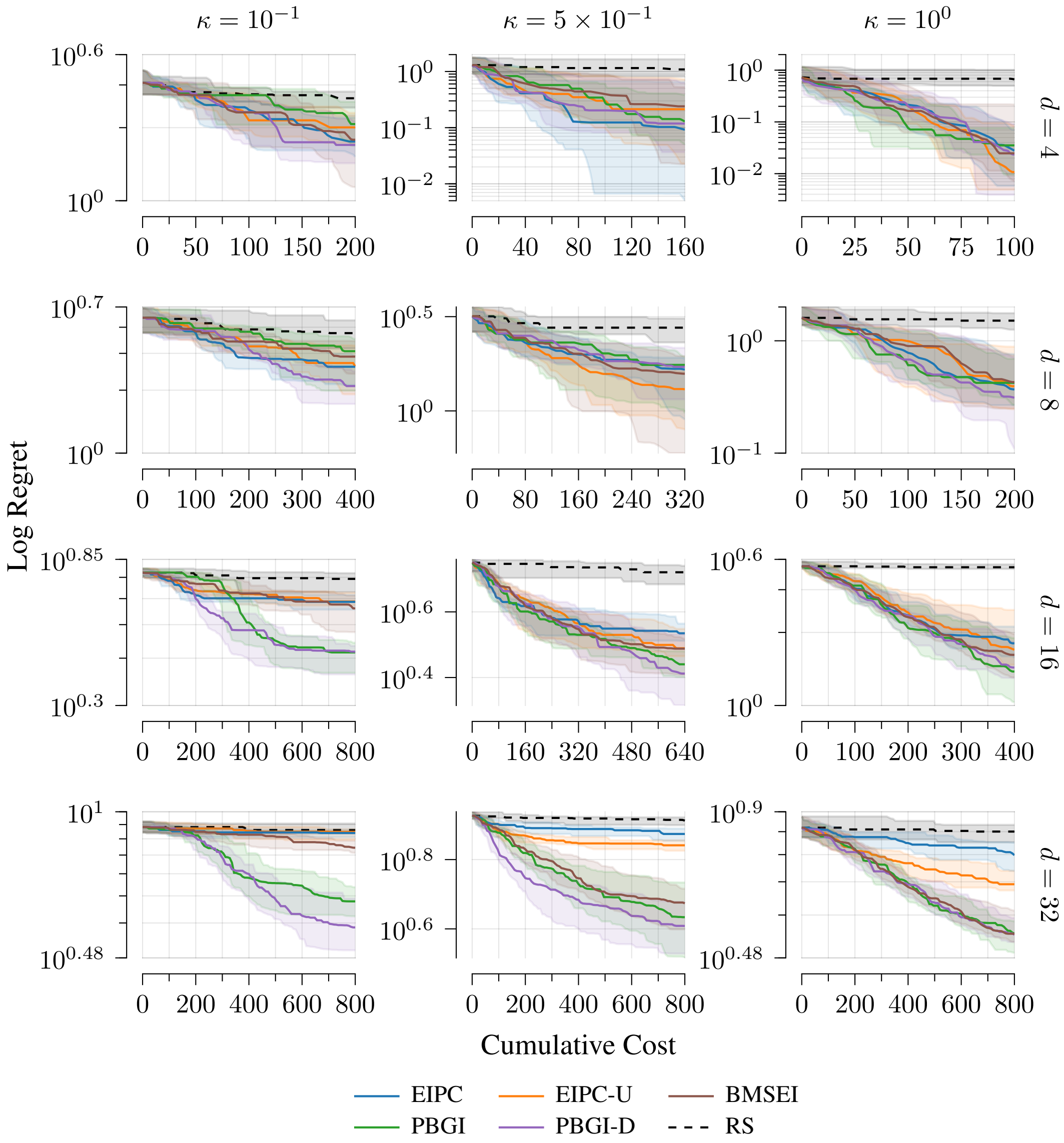

Figure 14: Comparison of Bayesian regret across different length scales and dimensions, with a Matérn-5/2 kernel, in the cost-aware setting. We see that overall behavior is similar, but the precise thresholds at which each example switches between the easy, medium-hard, and very hard regimes differ.

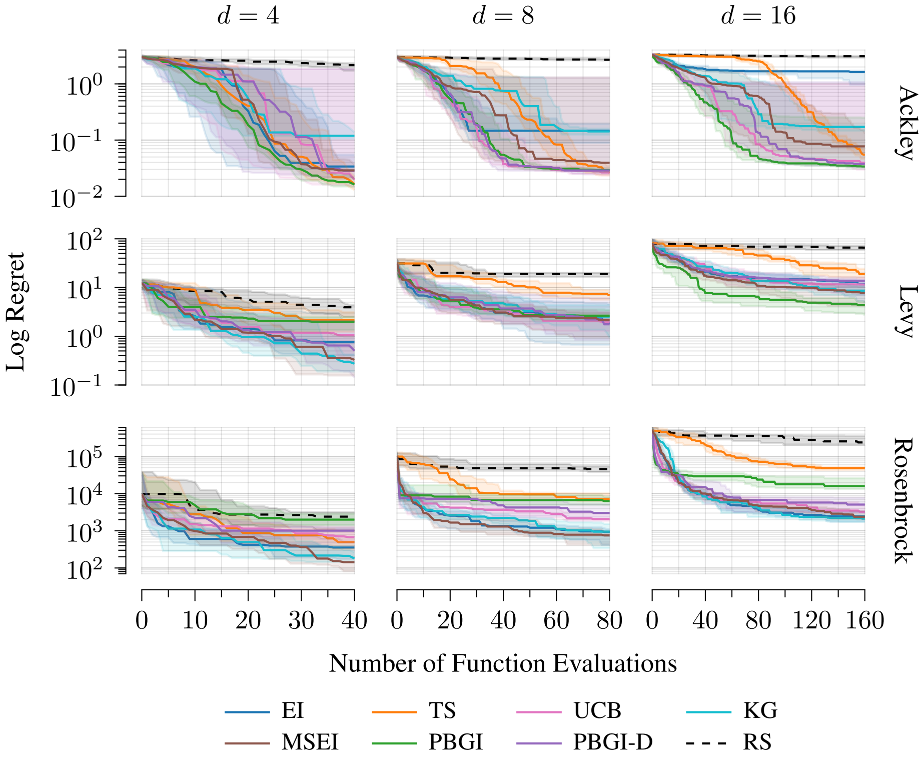

Figure 15: Comparison of regret for synthetic benchmark functions under different dimensions, in the uniform-cost setting. We see that all methods perform similarly for $d = 4$, with differences between the most-competitive methods emerging as dimension increases to $d = 8$ and $d = 16$.

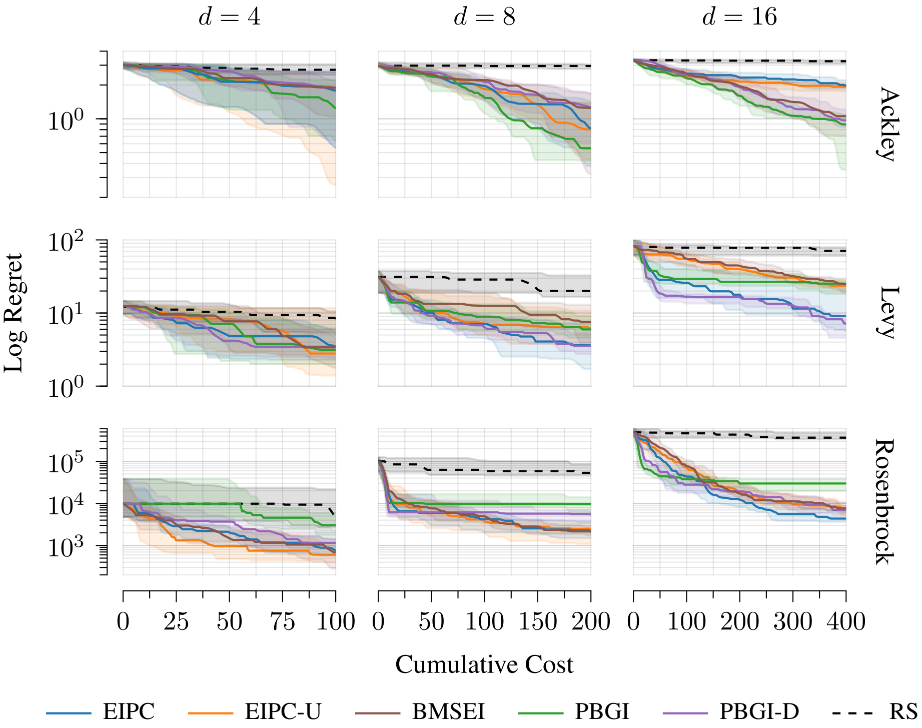

Figure 16: Comparison of regret for synthetic benchmark functions under different dimensions, in the cost-aware setting. We see that all methods perform similarly for $d = 4$, with differences between the most-competitive methods emerging as dimension increases to $d = 8$ and $d = 16$.