Appendix for Monte Carlo Tree Search in high-dimensional continuous spaces using Voronoi optimistic optimization with regret bounds

Beomjoon Kim[1], Kyungjae Lee[2], Sungbin Lim[3]*, Leslie Pack Kaelbling[1], and Tomás Lozano-Pérez[1]
[1]MIT Computer Science and Artificial Intelligence Laboratory
{beomjoon,lpk,tlp}@mit.edu,
[2]Seoul National University
kyungjae.lee@cpslab.snu.ac.kr,
[3]UNIST
sungbin@unist.ac.kr

Theoretical analysis

Regret bound for Voronoi Optimistic Optimization

First we present a more general assumption than A5 which makes it possible to consider a broader class of functions and prove the *probabilistic* regret bound for VOO.

Definition 1 (Nested Level Set)**.** *We say that f has **nested level sets** if there exist a finite index set $\mathcal{L}$ and a class of disjoint and connected open sets $\left\{\mathcal{O}^{(\ell)} : \ell \in \mathcal{L}\right\}$ which satisfies the following conditions:*

(i) $\mathcal{X}_\star$ consists of finite number of disjoint and connected components $\mathcal{X}_\star^{(\ell)}$ such that $\mathcal{X}_\star^{(\ell)} \subset \mathcal{O}^{(\ell)}$ for $\ell \in \mathcal{L}$,

(ii) for each $\ell \in \mathcal{L}$, there exists a nonnegative decreasing sequence $\{\epsilon_j\}_{j\in\mathbb{N}}$ with $\sum_{j=1}^{\infty} \epsilon_j < \infty$ and corresponding connected open sets $\mathcal{O}_{\epsilon_j}^{(\ell)} := \left\{x \in \mathcal{O}^{(\ell)} : d(x, \mathcal{X}_\star^{(\ell)}) < \epsilon_j\right\}$ such that $x \in \mathcal{O}_{\epsilon_{j+1}}^{(\ell)}, z \in \mathcal{O}_{\epsilon_j}^{(\ell)}$ implies $f(x) \geq f(z)$, and for any sequence $\{x_j\}_{j\in\mathbb{N}}$ with $d_j := d(x_j, \mathcal{X}_\star^{(\ell)}) > \epsilon_j$, and generated Voronoi cell C_j, it holds for every $j \in \mathbb{N}$

$$\max\left\{\frac{\mu(\mathcal{O}_{d_j}^{(\ell)} \setminus \mathcal{O}_{\epsilon_{j+1}}^{(\ell)} \cap \mathcal{U}_{x_j}^{(\ell)} \cap C_j)}{\mu(C_j)}, \bar{\mu}(\mathcal{O}_{d_j}^{(\ell)} \setminus \mathcal{O}_{\epsilon_{j+1}}^{(\ell)} \cap \mathcal{U}_{x_j}^{(\ell)})\right\} \leq 1 - e^{-\epsilon_j} \tag{1}$$

where $\mathcal{U}_{x_j}^{(\ell)} := \{x \in \mathcal{O}^{(\ell)} : f(x) < f(x_j)\}$.

Note that (1) is the key property which determines the probabilistic behavior of VOO algorithm. Intuitively, each $\mathcal{O}_{\epsilon_j}^{(\ell)}$ can be seen as a "level set" with respect to the component $\mathcal{X}_\star^{(\ell)}$, and ϵ_j defines how quickly the outer set $\mathcal{O}_{\epsilon_{j-1}}^{(\ell)}$ changes to $\mathcal{O}_{\epsilon_j}^{(\ell)}$ as we move toward $\mathcal{X}_\star^{(\ell)}$. ϵ_j can also be seen as the measure of asymmetric area between these two neighboring nested level sets.

From this definition, we can see that the right-hand side of (1) controls the probability for event when the best Voronoi cell differs with the optimal cell. Therefore, if we have n number of iterations, the probability that the best Voronoi cell and the optimal cell match is greater than

$$\prod_{j=1}^{n}\{1 - (1 - e^{-\epsilon_j})\} = \prod_{j=1}^{n} e^{-\epsilon_j} = e^{-\sum_{j=1}^{n}\epsilon_j} \to \Delta > 0 \tag{2}$$

where $\Delta := e^{-\sum_{j=1}^{\infty}\epsilon_j}$.

As we already mentioned in the main paper, A5 is a special case of Definition 1. If we assume f has nested level sets instead A5, then Theorem 1 holds with probability $\Delta > 0$. Due to the following lemma, we can take Δ arbitrarily near one hence Theorem 1 in the main paper holds almost surely (i.e. with probability 1) under A5.

Lemma 1. *If f satisfies A5, then f has nested level sets and satisfies (2) with arbitrarily small ϵ_j i.e. $\Delta \approx 1$.*

Proof. It is sufficient to show A5 satisfies the last property of nested level set assumption since the other statements hold if we take $\mathcal{O}^{(\ell)} = B_{\nu_\ell}$ and $\mathcal{L} = \{1, \ldots, k\}$. Take an arbitrary sequence $\{\epsilon_j : j \in \mathbb{N}\}$ with $\epsilon_1 = \nu_\ell$, $\sum_{j\in\mathbb{N}}\epsilon_j < \infty$ and define a sequence of open balls $B_{\epsilon_j} := \left\{x \in \mathcal{O}^{(\ell)} : d(x, x_\star) < \epsilon_j\right\}$. Then for any sequence $\{x_j\}_{j\in\mathbb{N}}$ with $d(x_j, x_\star) > \epsilon_j$, we can observe that $\mathcal{O}_{d_j}^{(\ell)} \cap \mathcal{U}_{x_j}^{(\ell)} = \emptyset$ since $d(x, x_\star) < d(x_j, x_\star)$ implies $f(x) \geq f(x_j)$. This implies the left-side of (1) vanishes since the measure of negligible set is zero. Therefore, we can take $\Delta \approx 1$ in (2) by taking a sequence ϵ_j which decays fast. The lemma is proved. □

*This work is done at Kakao Brain

Next we derive the upper-bound on the probability of sampling a point inside one of $\left\{O^{(\ell)} : \ell \in \mathcal{L}\right\}$ for the first time. Recall $\bar{\mu}_B(r) := \frac{\mu(B_r(\cdot))}{\mu(\mathcal{X})}$ and $\bar{\mu}(X) := \frac{\mu(X)}{\mu(\mathcal{X})}$ for a measurable subset $X \subset \mathcal{X}$. Define $\nu_{min} = \min_{\ell\in[k]} \nu_\ell$. In the following lemma, we assume $O^{(\ell)} = B_{\nu_\ell}(x_\star^{(\ell)})$, however, the statement generally holds if we use $\min_{\ell\in[k]} \bar{\mu}(O^{(\ell)})$ instead of $\bar{\mu}_B(\nu_{\min})$.

Lemma 2. *Define τ as the hitting time at which we have a sample inside the union of the ball $\bigcup_{\ell=1}^k B_{\nu_\ell}(x_\star^{(\ell)})$. We have*

$$\mathbb{P}(\tau = t) \le \{1 - \omega \cdot k\bar{\mu}_B(\nu_{min})\}^{t-1}$$

Proof. The hitting time τ is a random variable whose probability distribution follows a geometric distribution. Define $O := \bigcup_{\ell=1}^k B_{\nu_\ell}$ and C_i as the best Voronoi cell at iteration i. At any given time, the success probability of τ is

$$\begin{aligned} p_i &= \omega \cdot \bar{\mu}(O) + (1-\omega) \cdot \frac{\mu(O \cap C_i)}{\mu(C_i)} \\ &= \omega \cdot \bar{\mu}(O) + (1-\omega) \cdot \bar{\mu}(C_i)\bar{\mu}(O \cap C_i) \end{aligned}$$

and

$$1 - p_i = 1 - \omega \cdot \bar{\mu}(O) - (1-\omega) \cdot \bar{\mu}(C_i)\bar{\mu}(O \cap C_i)$$

Then our probability mass function becomes

$$\mathbb{P}(\tau = t) = p_t \prod_{i=1}^{t-1}(1 - p_i)$$

Denote $\bar{p}$ and $\underline{p}$ as upper and lower bound of p_i respectively. Then

$$\mathbb{P}(\tau = t) \le \bar{p}(1 - \underline{p})^{t-1}$$

Define $\nu_{max} = \max_{\ell\in[k]} \nu_\ell$. Since it is difficult to compute the intersection of the best cell and open balls $\left\{B_{\nu_\ell}\right\}$, we resort to the following:

$$\bar{p} = \omega \cdot k\bar{\mu}_B(\nu_{\max}) + (1-\omega)$$

and

$$\underline{p} = \omega \cdot k\bar{\mu}_B(\nu_{\min})$$

Then we get the desired result. □

Using this result, our strategy is to prove the regret bound by defining the *success event* as obtaining a sample from the Voronoi cell containing the optimal point, and failure event as otherwise, and then computing the upper-bound on the expected regret based on the probabilities of these events. This requires defining an upper-bound of a special function, which is done Lemma 3; we begin by introducing the special function.

Let $\mathfrak{F}_{\{a,b\}}^{\{c\}}(\cdot)$ be a *generalized hypergeometric function* which is defined by

$$\mathfrak{F}_{\{c\}}^{\{a,b\}}(z) := \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!} \tag{3}$$

where $(\alpha)_k = \alpha(\alpha+1)\cdots(\alpha+k-1)$. In general, (3) makes sense for $|z| < 1$. However, if a, b are negative integers then (3) is a finite sum hence one can define $\mathfrak{F}_{\{c\}}^{\{a,b\}}(z)$ for $|z| > 1$.

Lemma 3. *Let $n \in \mathbb{N}$ and $t, m \in \{0, \cdots, n\}$ be given and assume $p, q \in (0,1)$. Then for $n \ge t + m$,*

$$p^m(1-q)^t \mathfrak{F}_{\{1-m+n-t\}}^{\{-m,-t\}}\left(\frac{q(1-p)}{p(1-q)}\right) \le 1 \tag{4}$$

and for $n < t + m$,

$$p^{n-t}(1-q)^{n-m} \mathfrak{F}_{\{1+m+t-n\}}^{\{m-n,t-n\}}\left(\frac{q(1-p)}{p(1-q)}\right) \le 1 \tag{5}$$

Proof. One can find some well-known polynomial-type upper bounds of hypergeometric function from [2]. To prove this lemma rigorously, one can transform $\mathfrak{F}_{\{c\}}^{\{a,b\}}$ to Fox H function (see [5, (2.9.15)]) which is defined by Mellin-Barnes type integral: for $a, b, c \in \mathbb{C}, c \notin \{0, -1, -2, \cdots\}$

$$\frac{\Gamma(a)\Gamma(b)}{\Gamma(c)} \mathfrak{F}_{\{c\}}^{\{a,b\}}(z) = \frac{1}{2\pi i} \int_L \frac{\Gamma(s)\Gamma(a-s)\Gamma(b-s)}{\Gamma(c-s)} (-z)^{-s} ds$$

Since both $\mathfrak{F}^{\{-m,-t\}}_{\{1-m+n-t\}}, \mathfrak{F}^{\{m-n,t-n\}}_{\{1+m+t-n\}}$ are increasing functions, it is sufficient to show (4) and (5) holds for $p \to 0, q \to 1$. In this case, we can apply some asymptotic behaviors of Fox H-functions which are introduced in [6]. Therefore, for $n \geq t + m$,

$$p^m(1-q)^t \mathfrak{F}^{\{-m,-t\}}_{\{1-m+n-t\}}\left(\frac{q(1-p)}{p(1-q)}\right) \to 1$$

and for $n < t + m$

$$p^{n-t}(1-q)^{n-m} \mathfrak{F}^{\{m-n,t-n\}}_{\{1+m+t-n\}}\left(\frac{q(1-p)}{p(1-q)}\right) \to 1$$

The lemma is proved. □

Lemma 4. *Let*

$$\mathscr{S}(n,t,m,q,p) := \sum_{\ell=\max(0,m+t-n)}^{\min(t,m)} \binom{t}{\ell}\binom{n-t}{m-\ell} q^\ell (1-q)^{t-\ell} p^{m-\ell} (1-p)^{n-t-m+\ell} \tag{6}$$

Then for $t + m \leq n$

$$\mathscr{S}(n,t,m,q,p) \leq \binom{n-t}{m}(1-p)^{n-t-m}$$

and for $t + m \geq n + 1$

$$\mathscr{S}(n,t,m,q,p) \leq \binom{t}{m+t-n} q^{m+t-n}$$

Proof. Having (3) in mind, observe that if $m \leq n - t$ then (4) implies

$$\begin{aligned}
\mathscr{S}(n,t,m,q,p) &= \sum_{l=0}^{\min(t,m)} \binom{t}{l}\binom{n-t}{m-l} q^l (1-q)^{t-l} p^{m-l} (1-p)^{n-t-m+l} \\
&= (1-p)^{n-t-m} p^m (1-q)^t \sum_{l=0}^{\min(t,m)} \binom{t}{l}\binom{n-t}{m-l} q^l (1-q)^{-l} p^{-l} (1-p)^l \\
&= (1-p)^{n-t-m} p^m (1-q)^t \sum_{l=0}^{\min(t,m)} \frac{t!}{l!(t-l)!}\frac{(n-t)!}{(m-l)!(n-t-m+l)!} q^l (1-q)^{-l} p^{-l} (1-p)^l \\
&= \binom{n-t}{m}(1-p)^{n-t-m} p^m (1-q)^t \sum_{l=0}^{\min(t,m)} \frac{(-m)_l (-t)_l}{(1-m+n-t)_l}\frac{1}{(l!)}\left(\frac{q(1-p)}{p(1-q)}\right)^l \\
&\leq \binom{n-t}{m}(1-p)^{n-t-m} p^m (1-q)^t \mathfrak{F}^{\{-m,-t\}}_{\{1-m+n-t\}}\left(\frac{q(1-p)}{p(1-q)}\right) \leq \binom{n-t}{m}(1-p)^{n-t-m}
\end{aligned}$$

and if $t + m \geq n + 1$ then (5) implies

$$\begin{aligned}
\mathscr{S}(n,t,m,q,p) &= \sum_{l=m+t-n}^{\min(t,m)} \binom{t}{l}\binom{n-t}{m-l} q^l (1-q)^{t-l} p^{m-l} (1-p)^{n-t-m+l} \\
&= q^{m+t-n} p^{n-t} (1-q)^{n-m} \sum_{l=m+t-n}^{\min(t,m)} \binom{t}{l}\binom{n-t}{m-l} q^l (1-q)^{-l} p^{-l} (1-p)^l \\
&= q^{m+t-n} p^{n-t} (1-q)^{n-m} \sum_{k=0}^{n-\max(t,m)} \frac{(m-n)_k (t-n)_k}{(1+m+t-n)_k}\frac{1}{(k!)}\left(\frac{q(1-p)}{p(1-q)}\right)^k \ (k \triangleq l-m-t+n) \\
&\leq \binom{t}{m+t-n} q^{m+t-n} p^{n-t} (1-q)^{n-m} \mathfrak{F}^{\{m-n,t-n\}}_{\{1+m+t-n\}}\left(\frac{q(1-p)}{p(1-q)}\right) \leq \binom{t}{m+t-n} q^{m+t-n}
\end{aligned}$$

The lemma is proved. □

We now prove the main theorem, which we restate.

Theorem 1. *Let n be the total number of evaluations and $\frac{1-\lambda^{1/k}}{k\bar{\mu}_B(\nu_{min})+1-\bar{\mu}_B(\eta\cdot\lambda\delta_{max})} < \omega$ We have*

$$\mathcal{R}_n \leq L\delta_{max}C_1\left[\lambda^{1/k} + \omega(1-\bar{\mu}_B(\eta\cdot\lambda^n\delta_{max}))\right]^n + L\delta_{max}C_2[(1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})]^n \tag{7}$$

where C_1 and C_2 are constants independent of n as follows

$$C_1 := \frac{1}{1-\rho(\lambda^{1/k}+1-[1-\omega+\omega\bar{\mu}_B(\eta\cdot\lambda\delta_{max})])^{-1}},$$
$$\rho := 1-\omega k\bar{\mu}_B(\nu_{min}),$$

and

$$C_2 := \frac{\lambda^{-1/k}+1}{(\lambda^{-1/k}+1)-(1-\omega k\bar{\mu}_B(\nu_{min}))^{-1}}.$$

Proof. Define $M_i(n)$ as the random variable that corresponds to the number of times we sampled from the ith optimal Voronoi cell with n trials.

$$\begin{aligned}
\mathcal{R}_n &= f(x_\star) - \mathbb{E}_{x_{1:n}\sim \mathrm{voo}(f)}\left[\max_{i\in[n]} f(x_i)\right] \\
&\leq L\cdot\mathbb{E}_{x_{1:n}\sim\mathrm{voo}(f)}\left[\min_{i\in[n],j\in[k]} d(x_i, x_\star^{(j)})\right] \\
&\leq L\cdot\mathbb{E}_{x_{1:n}\sim\mathrm{voo}(f),\ M_{1:k_0}(n)}[\lambda^{\max\{M_1(n),\cdots,\ M_{k_0}(n)\}}\delta_{\max}]
\end{aligned}$$

where L is a constant of local smoothness assumption, $k_0 \leq k$ is the expected number of optimal cells that gets created when we run voo on f for n iterations. Now, because $\sum_{i=1}^{k_0} M_i(n) \leq \max\{M_1(n),\cdots,\ M_{k_0}(n)\}\cdot k_0$, we can write

$$\mathcal{R}_n \leq L\cdot\mathbb{E}_{x_{1:n}\sim\mathrm{voo}(f),\ M_{1:k_0}(n)}[\lambda^{\sum_{i=1}^{k_0} M_i(n)/k_0}\delta_{\max}] \quad (\because 0<\lambda<1)$$

We will write $k_0 = k$, and if $k_0 < k$, then set the remaining $k-k_0$ $M_i(n)$ to be zero. Now define $Y(n) := \sum_{i=1}^{k} M_i(n)$, which indicates the total number of times *any one of optimal cells* has been selected. Continuing from the above, we have

$$\begin{aligned}
\mathcal{R}_n &\leq L\cdot\mathbb{E}_{x_{1:n}\sim\mathrm{voo}(f),\ M_{1:k_0}(n)}[\lambda^{\sum_{i=1}^{k_0} M_i(n)/k_0}\delta_{\max}] \\
&\leq L\cdot\mathbb{E}_{x_{1:n}\sim\mathrm{voo}(f),\ Y(n)}[\lambda^{Y(n)/k}\delta_{\max}]
\end{aligned}$$

Define *hitting time* τ as the iteration at which we have a point inside the ball $B_\nu^{(i)}(x_\star)$ from A3. Denote $m_{<t}$ as the number of samples inside the optimal Voronoi cell before hitting time $m_{\geq t}$ after the hitting time. Denote $C(j)$ as the best cell, and $C_\ell^\star(j)$ as the optimal Voronoi cell containing the ℓ^{th} optimum, after $j-1$ points are evaluated in the optimal cell. Further, denote the union of optimal cells as $C_U^\star(j) = \bigcup_{\ell=1}^k C_\ell^\star(j)$. Also, denote q_j as the probability of sampling a point from the optimal Voronoi cell when the total number of points in all of the optimal cells is $j-1$ before the hitting time, where $j\in\{1,\ldots,m_{<t}\}$, and similarly for p_j, but after the hitting time, so $j\in\{1,\ldots,m_{\geq t}\}$.

Then, we compute the success probability.

$$q_j := (1-\omega)\mathbb{P}(C(j)\subset C_U^\star(j))) + \omega\bar{\mu}(C_U^\star(j)), \quad j\in\{1,\ldots,m_{<t}\}$$

and

$$p_j := (1-\omega) + \omega\bar{\mu}(C_U^\star(j)), \quad j\in\{1,\ldots,m_{\geq t}\}$$

Thanks to A3, after the hitting time, $\mathbb{P}(x_j\in C_U^\star(j)) = 1$ since the best Voronoi cell is always included in the union of the optimal cells after hitting the ball $B_\nu(x_\star)$. We wish to compute the upper bound of $\mathbb{P}_Y$, using the definition of q_i and p_j. By Lemma 2 with $\rho := 1-\omega k\bar{\mu}_B(\nu_{min})$,

$$\mathbb{P}_{Y(n)}(m) = \sum_{t=0}^{n}\sum_{m=0}^{n}\mathbb{P}(\tau=t)\mathbb{P}\left(\sum_{i=1}^{n} X_i = m\middle|\tau=t\right) \leq \sum_{t=0}^{n}\rho^t\sum_{m=0}^{n}\mathbb{P}\left(\sum_{i=1}^{n} X_i = m\middle|\tau=t\right)$$

where X_i indicates a success of which the ith point is sampled from $C_U^\star(j)$, and $\mathbb{P}\left(\sum_{i=1}^n X_i = m|\tau=t\right)$ can be written by

$$\sum_{m_{<t}=\max(0,m+t-n)}^{\min(t,m)}\left(\sum_{\substack{\alpha_1+\cdots+\alpha_{m_{<t}}=t-m_{<t}\\ \beta_1+\cdots\beta_{m_{\geq t}}=n-t+m_{\geq t}}}\prod_{i=1}^{m_{<t}}(1-q_i)^{\alpha_i}q_i\prod_{j=1}^{m_{\geq t}}(1-p_j)^{\beta_j}p_j\right) \tag{8}$$

Here each α_i denotes the number of failures between $(i-1)$-th success and i-th success before t and each β_j indicates the number of failures between $(j-1)$-th success and j-th success after t. Note that $\alpha_1, \ldots, \alpha_{m<t}, \beta_1, \ldots, \beta_{m\geq t}$ are non-negative integers. Given $t, m \leq n$, let us abbreviate (8) to $\mathscr{G}(q_1, \ldots, q_n; p_1, \ldots, p_n)$ and recall (6) to define a continuous function $\mathscr{S}(x, y) := \mathscr{S}(n, t, m, x, y)$ for $x, y \in [0, 1]$. Here we omit the notation n, t, m in $\mathscr{S}$ and $\mathscr{G}$ for notational convenience.

To compute the upper bound of this probability, we first define probabilities $\bar{q}$ and $\bar{p}$, $\underline{q}$ and $\underline{p}$, where

$$\underline{q} < \min\{q_1, \ldots, q_n\}, \quad \bar{q} > \max\{q_1, \ldots, q_n\}$$

$$\underline{p} < \min\{p_1, \ldots, p_n\}, \quad \bar{p} > \max\{p_1, \ldots, p_n\}$$

Now because any cell begins with the diameter of δ_{max}, using the Shrinkage ratio assumption (A4), we can see that any Voronoi cell evaluated $i-1$ times would have an expected diameter less than or equal to $\lambda^{i-1} \cdot \delta_{\max}$. Also, the Well-shaped Voronoi cell assumption (A5) tells us that such cell contains a ball of radius $\eta \cdot \lambda^{i-1} \cdot \delta_{\max}$, for some $\eta > 0$, centered at the point that defines the cell. Thus, we can write

$$\begin{aligned} \bar{q} &= 1 \\ \underline{p} &= (1-\omega) + \omega \cdot \bar{\mu}_B(\eta \cdot \lambda^n \delta_{max}) \end{aligned} \tag{9}$$

where we omitted the center point of the ball for brevity. Then observe that $\mathscr{S}(q_i, p_j) = \mathscr{G}(q_i, \ldots, q_i; p_j, \ldots, p_j)$ for each i, j and

$$\min_{i,j} \mathscr{S}(q_i, p_j) \leq \mathscr{G}(q_1, \ldots, q_n; p_1, \ldots, p_n) \leq \max_{i,j} \mathscr{S}(q_i, p_j) \tag{10}$$

Since $\mathscr{S}$ is continuous and $[\underline{p}, \bar{p}] \times [\underline{q}, \bar{q}]$ is connected, its image via $\mathscr{S}$ is also connected (see [9, Theorem 4.22]). This and (10) imply there exist $\tilde{q} = \tilde{q}(n, t, m)$, $\tilde{p} = \tilde{p}(n, t, m)$ such that

$$\underline{q} \leq \tilde{q} \leq \bar{q}, \quad \underline{p} \leq \tilde{p} \leq \bar{p}$$

and

$$\mathscr{G}(q_1, \ldots, q_n; p_1, \ldots, p_n) = \mathscr{S}(\tilde{q}, \tilde{p})$$

Therefore

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i = m \middle| \tau = t\right) = \mathscr{S}(n, t, m, \tilde{q}, \tilde{p})$$

Then, by Lemma (4), for $m + t \leq n$

$$\mathscr{S}(n, t, m, \tilde{q}, \tilde{p}) \leq \binom{n-t}{m}(1-\tilde{p})^{n-t-m} \leq \binom{n-t}{m}\left(1-\underline{p}\right)^{n-t-m}$$

and for $m + t > n$

$$\mathscr{S}(n, t, m, \tilde{q}, \tilde{p}) \leq \binom{t}{m+t-n}(\tilde{q})^{m+t-n} \leq \binom{t}{m+t-n}(\overline{q})^{m+t-n}$$

Therefore,

$$R_n \leq L\delta_{max} \sum_{t=0}^{n}\sum_{m=0}^{n} \rho^t \lambda^{m/k} \mathbb{P}\left(\sum_{i=1}^{n} X_i = m \middle| \tau = t\right) \leq L\delta_{max}(\mathfrak{I}_1 + \mathfrak{I}_2)$$

where

$$\mathfrak{I}_1 := \sum_{t=0}^{n}\sum_{m=0}^{n-t} \rho^t \lambda^{m/k}\binom{n-t}{m}\left(1-\underline{p}\right)^{n-t-m}, \quad \mathfrak{I}_2 := \sum_{t=0}^{n}\sum_{m=n-t+1}^{n} \rho^t \lambda^{m/k}\binom{t}{m+t-n}(\overline{q})^{m+t-n}$$

These are

$$\begin{aligned}
\mathfrak{J}_1 &= \sum_{t=0}^{n}\sum_{m=0}^{n-t}\rho^t\lambda^{m/k}\binom{n-t}{m}(1-\underline{p})^{n-t-m} \\
&= \sum_{t=0}^{n}\rho^t\left(\sum_{m=0}^{n-t}\binom{n-t}{m}\lambda^{m/k}\left(1-\underline{p}\right)^{n-t-m}\right) \\
&= \sum_{t=0}^{n}\rho^t\left(\lambda^{1/k}+1-\underline{p}\right)^{n-t} \\
&= \frac{(\lambda^{1/k}+1-\underline{p})^n-\rho^{n+1}(\lambda^{1/k}+1-\underline{p})^{-1}}{1-\rho\left(\lambda^{1/k}+1-\underline{p}\right)^{-1}} \\
&\leq \frac{\left(\lambda^{1/k}+1-\underline{p}\right)^n}{1-\rho\left(\lambda^{1/k}+1-\underline{p}\right)^{-1}} \\
&\leq C_1\cdot\left(\lambda^{1/k}+1-\underline{p}\right)^n
\end{aligned}$$

where $C_1 \triangleq \sup_{\underline{p}}\left(1-\rho\left(\lambda^{1/k}+1-\underline{p}\right)\right)^{-1}$. Note that $\underline{p}$ decreases as $n\to\infty$ due to (9), therefore C_1 is determined by the value of $\underline{p}$ from (9) for $n=1$. For this $\underline{p}$, C_1 is positive if $\rho<\lambda^{1/k}+1-\underline{p}$ holds. We now prove the range of ω such that this is true. We need

$$\begin{aligned}
1-\omega k\bar{\mu}_B(\nu_{min}) &< \lambda^{1/k}+1-[1-\omega+\omega\bar{\mu}_B(\eta\cdot\lambda\delta_{max})] \\
&= \lambda^{1/k}+\omega(1-\bar{\mu}_B(\eta\cdot\lambda\delta_{max}))
\end{aligned}$$

Note that the above holds if

$$\frac{1-\lambda^{1/k}}{k\bar{\mu}_B(\nu_{min})+1-\bar{\mu}_B(\eta\cdot\lambda\delta_{max})} < \omega \tag{11}$$

We now compute $\mathfrak{J}_2$.

$$\begin{aligned}
\mathfrak{J}_2 &= \sum_{t=0}^{n}\sum_{m=n-t+1}^{n}\rho^t\lambda^{m/k}\binom{t}{m+t-n}\bar{q}^{m+t-n} \\
&= \sum_{t=0}^{n}\sum_{m=1}^{t}\rho^t\binom{t}{m}(\lambda^{1/k})^{n-(t-m)}\bar{q}^m \\
&\leq \lambda^{n/k}\sum_{t=0}^{n}\rho^t\left(\lambda^{-1/k}+\bar{q}\right)^t \\
&= \lambda^{n/k}\frac{(\rho(\lambda^{-1/k}+\bar{q}))^{n+1}-1}{(\rho(\lambda^{-1/k}+\bar{q}))-1} \\
&= \frac{1}{\rho(\lambda^{-1/k}+\bar{q})-1}\left(\rho(\lambda^{-1/k}+\bar{q})(\rho(1+\bar{q}\lambda^{1/k}))^n-\lambda^{n/k}\right) \\
&\leq C_2\cdot(\rho(1+\bar{q}\lambda^{1/k}))^n = C_2\cdot(\rho(1+\lambda^{1/k}))^n
\end{aligned}$$

where $C_2 \triangleq \frac{\rho(\lambda^{-1/k}+\bar{q})}{\rho(\lambda^{-1/k}+\bar{q})-1}$.

□

Corollary 1. *If*

$$\frac{\lambda^{1/k}}{(1+\lambda^{1/k})k\bar{\mu}_B(\nu_{min})} < \omega < 1-\lambda^{1/k}, \quad \frac{\lambda^{1/k}}{1-\lambda^{2/k}} < k\bar{\mu}_B(\nu_{min}) \tag{12}$$

then $\lim_{n\to\infty}\mathcal{R}_n = 0$.

Proof. Note that the second condition of (12) guarantees that the range of ω is non-empty. Due to the proof of Theorem 1, we have the following inequalities:

$$\mathfrak{I}_1 \leq \frac{(\lambda^{1/k} + 1 - \underline{p})^n - \rho^{n+1}(\lambda^{1/k} + 1 - \underline{p})^{-1}}{1 - \rho\left(\lambda^{1/k} + 1 - \underline{p}\right)^{-1}} \tag{13}$$

$$\mathfrak{I}_2 \leq \frac{1}{\rho(\lambda^{-1/k} + \bar{q}) - 1}\left(\rho(\lambda^{-1/k} + \bar{q})(\rho(1 + \bar{q}\lambda^{1/k}))^n - \lambda^{n/k}\right) \tag{14}$$

To get the desired result, we prove that both (13) and (14) converge to zero under (12) as $n \to \infty$. Observe that $\frac{\lambda^{1/k}}{(1+\lambda^{1/k})k\bar{\mu}_B(\nu_{min})} < \omega$ implies $(1 - \omega k\bar{\mu}_B(\nu_{min}))(1 + \lambda^{1/k}) < 1$. Thus,

$$0 < \rho(1 + \bar{q}\lambda^{1/k}) \leq \rho(1 + \lambda^{1/k}) = (1 - \omega k\bar{\mu}_B(\nu_{min}))(1 + \lambda^{1/k}) < 1$$

Hence the right-hand side of (14) converges to zero as $n \to \infty$. Indeed, $\lambda \in (0, 1)$ and the above inequality imply

$$\lim_{n\to\infty} \mathfrak{I}_2 \leq \frac{1}{\rho(\lambda^{-1/k} + \bar{q}) - 1} \lim_{n\to\infty}\left(\rho(\lambda^{-1/k} + \bar{q})(\rho(1 + \bar{q}\lambda^{1/k}))^n - \lambda^{n/k}\right) = 0$$

Now we prove (13) also converges to zero as $n \to \infty$. Since $\omega < 1 - \lambda^{1/k}$, there exists a sufficiently small $\varepsilon > 0$ such that

$$\omega < 1 - \lambda^{1/k} - \varepsilon \tag{15}$$

Then observe that

$$\begin{aligned}
\lambda^{1/k} + 1 - \underline{p} &= \lambda^{1/k} + \omega(1 - \bar{\mu}_B(\eta \cdot \lambda^n \delta_{max})) \\
&< \lambda^{1/k} + (1 - \lambda^{1/k} - \varepsilon)(1 - \bar{\mu}_B(\eta \cdot \lambda^n \delta_{max})) \\
&\leq \lambda^{1/k} + (1 - \lambda^{1/k} - \varepsilon) = 1 - \varepsilon
\end{aligned}$$

Therefore, we have $(\lambda^{1/k} + 1 - \underline{p})^n < (1 - \varepsilon)^n$ holds for any n. This and $\rho \in (0, 1)$ implies

$$\lim_{n\to\infty} \mathfrak{I}_1 \leq \frac{1}{1 - \rho\left(\lambda^{1/k} + 1 - \underline{p}\right)^{-1}} \lim_{n\to\infty}\left[(\lambda^{1/k} + 1 - \underline{p})^n - \rho^{n+1}(\lambda^{1/k} + 1 - \underline{p})^{-1}\right] = 0$$

Consequently

$$\lim_{n\to\infty} \mathcal{R}_n \leq \lim_{n\to\infty} (\mathfrak{I}_1 + \mathfrak{I}_2) = 0$$

The corollary is proved. □

Voronoi Optimistic Optimization applied to Trees

Define the *state-action value function* at depth h as

$$Q^{(h)}(s, u) = r(s, u) + \gamma V_\star^{(h+1)}(T(s, u))$$

the *value function* as

$$V_\star^{(h)} = \max_{u \in U} Q^{(h)}(s, u)$$

the *estimated* state-action value function at depth h as

$$\hat{Q}^{(h)}(s, u) = r(s, u) + \gamma \hat{V}_{N_r(h+1)}^{(h+1)}(T(s, u))$$

where $N_r(h + 1)$ denotes the number of the times the transition $T(s, u)$ has been evaluated, and the *estimated value function* as

$$\hat{V}_{N_r(h+1)}^{(h+1)}(s) = \max_{u_{1:N_r(h+1)} \sim voo(\hat{Q}(s,\cdot))} \hat{Q}^{(h+1)}(s, u)$$

Further, denote the optimal estimated value function as $\hat{V}_\star^{(h)} = \max_{u \in U} \hat{Q}^{(h)}(s, u)$. We will omit the depth specification for these functions when the context is clear.

Define the regret of a node in height h as

$$V_\star^{(h)}(s) - \hat{V}_{N_r(h)}^{(h)}(s)$$

Our objective is to compute the total number of simulations, N_{iter}, required to upper-bound the regret of the value function at the root node to an arbitrary small number. Our strategy is to first compute the number of actions that needs to be sampled to upper-bound the regret at the leaf node, and then propagate the regret to the root. We have the following lemma that propagates the regret from depth $h + 1$ to h.

Lemma 5. *Suppose that at each node at height $h+1$ its regret is $\zeta(h+1)$, $\zeta(h+1) > 0$, and the range for ω hold as in Theorem 1. Then, the regret of a node at depth h is at most $\zeta(h)$ for some $\zeta(h) \geq \zeta(h+1)$ if*

$$N_r(h) \geq \log\left(\frac{\zeta(h) - \gamma\zeta(h+1)}{2L\delta_{max}C_{\max}}\right)\min\left(G_{\lambda,\omega}, K_{\lambda,\omega,\nu}\right)$$

where $G_{\lambda,\omega} = (\log\left(\lambda^{1/k} + \omega\right))^{-1}$ *and*, $K_{\lambda,\omega,\nu} = (\log([(1 - \omega k\bar{\mu}_B(\nu_{min})) \cdot (1 + \lambda^{1/k})]))^{-1}$

Proof. The regret of the value function can be written as

$$V_\star^{(h)}(s) - \hat{V}_{N_r(h)}^{(h)}(s) = \left(V_\star^{(h)}(s) - \hat{V}_\star^{(h)}(s)\right) + \left(\hat{V}_\star^{(h)}(s) - \hat{V}_{N_r(h)}^{(h)}(s)\right)$$

First consider $\hat{V}_\star^{(h)}(s) - \hat{V}_{N_r(h)}^{(h)}(s)$. From the definition, this is the quantity is upper-bounded by the regret-bound of voo that optimizes $\hat{Q}^{(h)}(s,\cdot)$. Thus, we have

$$\hat{V}_\star^{(h)}(s) - \hat{V}_{N_r(h)}^{(h)}(s) \leq \mathcal{R}_{N_r(h)}$$

Now, we consider the other term. Define $u_*(h) = \arg\max_{u\in U} Q^{(h)}(s,u)$. We have

$$\begin{aligned}
V_\star^{(h)}(s) - \hat{V}_\star^{(h)}(s) &= \max_{u\in U} Q^{(h)}(s,u) - \max_{u\in U}\hat{Q}^{(h)}(s,u) \\
&\leq Q^{(h)}(s,u_*(h)) - \hat{Q}^{(h)}(s,u_*(h)) \\
&= \gamma\cdot\left[V_\star^{(h+1)}(T(s,u_*(h))) - \hat{V}_{N_r(h+1)}^{(h+1)}(T(s,u_*(h)))\right] \\
&= \gamma\cdot\zeta(h+1)
\end{aligned}$$

Then, our total regret is

$$V_\star^{(h)}(s) - \hat{V}_{N_r(h)}^{(h)}(s) \leq \mathcal{R}_{N_r(h)} + \gamma\cdot\zeta(h+1) \tag{16}$$

We wish to compute $N_r(h)$ that keeps the regret to be at most $\zeta(h)$. We have

$$\begin{aligned}
\zeta(h) &\geq \mathcal{R}_{N_r(h)} + \gamma\cdot\zeta(h+1) \\
\zeta(h) - \gamma\zeta(h+1) &\geq \mathcal{R}_{N_r(h)}
\end{aligned}$$

The regret of voo is

$$\begin{aligned}
\mathcal{R}_n &= L\delta_{max}C_1\left[\lambda^{1/k} + \omega(1 - \bar{\mu}_B(\eta\cdot\lambda^n\delta_{max}))\right]^n + L\delta_{max}C_2[(1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})]^n \\
&\leq L\delta_{max}C_{\max}\left(\left[\lambda^{1/k} + \omega(1 - \bar{\mu}_B(\eta\cdot\lambda^n\delta_{max}))\right]^n + [(1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})]^n\right) \\
&\leq L\delta_{max}C_{\max}\cdot 2\cdot\max\left([\lambda^{1/k} + \omega(1 - \bar{\mu}_B(\eta\cdot\lambda^n\delta_{max}))]^n, [(1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})]^n\right) \\
&\leq L\delta_{max}C_{\max}\cdot 2\cdot\max\left([\lambda^{1/k} + \omega(1 - \bar{\mu}_B(\eta\cdot\lambda^n\delta_{max}))]^n, [(1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})]^n\right)
\end{aligned}$$

We will analyze each possibility. Suppose that the first argument of the max-operator was the maximum. Then,

$$\begin{aligned}
\zeta(h) - \gamma\zeta(h+1) &\geq C_{\max}\cdot 2\cdot[\lambda^{1/k} + \omega]^{n_1} \\
n_1 &\geq \frac{\log\left((\zeta(h) - \gamma\zeta(h+1))/2L\delta_{max}C_{\max}\right)}{\log\left([\lambda^{1/k} + \omega]\right)}
\end{aligned}$$

where the last inequality was flipped because the divisor was negative by our range of ω. Denote $G_{\lambda,\omega} = (\log\left(\lambda^{1/k} + \omega\right))^{-1}$ to obtain

$$n_1 \geq G_{\lambda,\omega}\cdot\log\left((\zeta(h) - \gamma\zeta(h+1))/2L\delta_{max}C_{\max}\right)$$

Now, we suppose that the second argument of the max operator was the maximum. Then,

$$\begin{aligned}
\zeta(h) - \gamma\zeta(h+1) &\geq LC_{max}\cdot 2\cdot[(1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})]^{n_2} \\
n_2 &\geq \frac{\log((\zeta(h) - \gamma\zeta(h+1))/(L\delta_{max}2C_{max}))}{\log([(1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})])} \\
n_2 &\geq K_{\nu,\omega,\lambda}\cdot\log((\zeta(h) - \gamma\zeta(h+1))/(L\delta_{max}2C_{max}))
\end{aligned}$$

where $K_{\nu,\omega,\lambda} = (\log([((1-\omega k\bar{\mu}_B(\nu_{min}))\cdot(1+\lambda^{1/k})])))^{-1}$. This gives

$$\begin{aligned} N_r(h) &= \max(n_1, n_2) \\ &= \log\left(\frac{\zeta(h)-\gamma\zeta(h+1)}{2L\delta_{max}C_{\max}}\right)\min\left(G_{\lambda,\omega}, K_{\lambda,\omega,\nu}\right) \end{aligned}$$

□

We now prove the main theorem.

Theorem 2. *Define $C_{max} = max\{C_1, C_2\}$. Given a decreasing sequence $\zeta(h)$ with respect to h, $h \in \{0\cdots H-1\}$, $\zeta(h) > 0$ and the range of ω as in Theorem 1, if $N_{iter} = \prod_{h=0}^{H-1} N_r(h)$ is used, where*

$$N_r(h) \geq \log\left(\frac{\zeta(h)-\gamma\zeta(h+1)}{2L\delta_{max}C_{max}}\right)\cdot\min(G_{\lambda,\omega}, K_{\nu,\omega,\lambda})$$

$G_{\lambda,\omega} = (\log\left(\lambda^{1/k}+\omega\right))^{-1}$, $K_{\nu,\omega,\lambda} = (\log([(1-\omega k\bar{\mu}_B(\nu_{min}))(1+\lambda^{1/k})]))^{-1}$, then for any state s traversed in the search tree we have

$$V_\star^{(h)}(s) - \hat{V}^{(h)}_{N_r(h)}(s) \leq \zeta(h) \quad \forall h \in \{0,\cdots,H-1\}$$

Proof. We use a mathematical induction on h. Suppose that $h = H-1$, and that we use $N_r(H-1)$ with $\zeta(H) = 0$ as the next state does not exist at the end of the planning horizon. Since the value function equals the reward function at this depth, this means

$$V_\star^{(H-1)}(s) - \hat{V}^{(H-1)}_{N_r(H-1)}(s) = \mathcal{R}_{N_r(H-1)}$$

The number of evaluations is at least

$$N_r(H-1) \geq \log\left(\frac{\zeta(H-1)}{2L\delta_{max}C_{max}}\right)\cdot\min(G_{\lambda,\omega}, K_{\nu,\omega,\lambda})$$

Substituting this to the regret-bound of voo, we get $R_{N_r(H-1)} \leq \zeta(H-1)$.

Now we assume that if we have used $N_r(H-1),\cdots,N_r(h+1)$, then the regrets at these depths are $\zeta(H-1),\cdots,\zeta(h+1)$, respectively. We now show the regret at depth h is ζ if we use $N_r(h)$ evaluations. Define $u_*(h) = \arg\max_{u\in U} Q^{(h)}(s,u)$.

$$\begin{aligned} V_\star^{(h)} - \hat{V}_\star^{(h)} &= \max_{u\in U} Q^{(h)}(s,u) - \max_{u\in U}\hat{Q}^{(h)}(s,u) \\ &\leq \max_{u\in U} Q^{(h)}(s,u) - \hat{Q}^{(h)}(s,u_*(h)) \\ &= \gamma\cdot\left[V_\star^{(h+1)}(T(s,u_*(h))) - \hat{V}^{(h+1)}_{N_r(h+1)}(T(s,u_*(h)))\right] \\ &\leq \gamma\cdot\zeta(h+1) \end{aligned}$$

Using the assumption, and applying Eqn (16), we have

$$\begin{aligned} V_\star^{(h)} - \hat{V}^{(h)}_{N_r(h)} &= \hat{V}_\star^{(h)} - \hat{V}^{(h)}_{N_r(h)} + V_\star^{(h)} - \hat{V}_\star^{(h)} \\ &= \mathcal{R}_{N_{r(h)}} + \gamma\cdot\zeta(h+1) \end{aligned}$$

Now, applying Lemma 5 we obtain

$$N_r(h) \geq \log\left(\frac{\zeta(h)-\gamma\zeta(h+1)}{2L\delta_{max}C_{max}}\right)\cdot\min(G_{\lambda,\omega}, K_{\nu,\omega,\lambda})$$

At each node in each height of the tree, we sample $N_r(h)$ number of actions to ensure that its regret is $\zeta(h)$. Thus, we need at least $\prod_{h=0}^{H-1} N_r(h)$ number of simulations to achieve the regret of $\zeta(h)$ at each node in the search tree. □

Remark 1. *If we set $\zeta(h)$ to be constant, i.e., $\zeta(h) := \zeta$. Then, $N_r(h)$ becomes a constant as follows.*

$$N_r(h) = \log\left(\frac{\zeta(1-\gamma)}{2L\delta_{max}C_{max}}\right)\cdot\min(G_{\lambda,\omega}, K_{\nu,\omega,\lambda}).$$

Let N_r be a constant $N_r(h)$. If $N_{iter} = N_r^H$ is used, for any state s traversed in the search tree we have

$$V_\star^{(h)}(s) - \hat{V}^{(h)}_{N_r(h)}(s) \leq \zeta(h) \quad \forall h \in \{0,\cdots,H-1\}$$

Remark 2. *Suppose that $\gamma = 1$. Let us define $\zeta(h) := \zeta\cdot(1-h/H)$ for $\zeta < 1$. Then, $N_r(h)$ becomes*

$$N_r(h) \geq \log\left(\frac{\zeta}{2HL\delta_{max}C_{max}}\right)\cdot\max(G_{\lambda,\omega}, K_{\nu,\omega,\lambda})$$

Thus, $N_r(h)$ is a constant and $N_{iter} = N_r^H$. If $N_{iter} = N_r^H$ is used, for any state s traversed in the search tree we have

$$V_\star^{(h)}(s) - \hat{V}^{(h)}_{N_r(h)}(s) \leq \zeta(h) \quad \forall h \in \{0,\cdots,H-1\}$$

Details of experiments

The budgeted-black-box function optimization problems

In this section, we describe the test functions, algorithm-specific implementation choices for the benchmarks, exploration parameters used, and their results. We begin with the implementation choices. For SOO and DOO, we use a binary tree to represent the partitions, and use the heuristic suggested in [8] that cuts the largest dimension when constructing a partition. We use the center of a cell as a representative point. To implement GP-UCB, we used the GPy library [4] for implementing GP. To obtain the global optimum of the acquisition function, we uniform-randomly sample the points at 10000 different locations, evaluate their acquisition function values, choose the point that has the highest value, and used gradient-based optimization to further optimize the acquisition function. For PW-UCT, we used the uniform action sampler because it always has a non-zero probability of sampling an optimal action, which satisfies the assumption in [1].

There are various inputs to the optimization algorithms that are summarized in Table 1. For DOO and VOO, we always use the same semi-metric in all problems which are defined in the subsequent sections. For GP-UCB, we use the radial basis function kernel with the initial variance of 20, and then use empirical Bayes to update the hyper-parameters after each evaluation of the objective function. For exploration parameters ζ, C, and ω, we tried various values, and report the best one. Table 1 describes the set of inputs and assumptions on the benchmarks.

	GP-UCB	SOO	DOO	VOO (ours)	HOO
Inputs	$K(\cdot,\cdot), \zeta$	None	$d(\cdot,\cdot), C$	$d(\cdot,\cdot), \omega$	$d(\cdot,\cdot), C$
Assumption on f	$f \sim \mathrm{GP}(\mu, \Sigma)$	Local smoothness	Local smoothness	Local smoothness	Weakly Lipschitz

Table 1: A list of inputs to each black-box function optimization algorithm, and their assumptions on the objective function f. K denotes the kernel function used to compute the covariance matrix Σ for GP, μ denotes the mean of GP, and ζ denotes the exploration parameter in GP-UCB. C denotes the Lipschitz constant of f, ω denotes the exploration probability for VOO, and $d(\cdot,\cdot)$ denotes the semi-metric defined on the search space.

Objective function definitions We now give the definitions of the objective functions used.

$$f_{\text{griewank}}(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad \mathcal{X} = [-600, 600]^N$$

$$f_{\text{rastrigin}}(x) = 10N + \sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i), \quad \mathcal{X} = [-5.12, 5.12]^N$$

$$f_{\text{shekel}}(x) = \sum_{i=1}^{M} \frac{1}{c_i + \sum_{j=1}^{N}(x_j - a_{ij})^2}, \quad \mathcal{X} = [-500, 500]^N$$

where N denotes the number of dimensions of the search space, M denotes the number of optimums for the Shekel function, c_i and a_{ij} denotes the parameters of the Shekel function that determines the locations of the optimums. We set $M = 10$ for all dimensions for the Shekel function.

Results on different hyperparameters Each algorithm has associated parameters that are summarized in Table 1. Here, we report the values of exploration parameters ζ, C, and ω that we tested in our experiments, and their results. Note that SOO does not have an exploration parameter. We report the optimum obtained averaged over 10 random seeds. The Avg column indicates the average of $\max_{x_{1:t}} f(x_t)$, for varying t . For DOO and VOO, they both used Euclidean distance metric as $d(\cdot,\cdot)$. The plots reported in the main paper is drawn with the best exploration parameter out of all the ones tried.

The Griewank function

ζ	Avg	95% CI
0.1	-0.154	0.05
1.0	-0.152	0.04
5.0	-0.177	0.06
10.0	-0.168	0.04
30.0	-0.110	0.03

ζ	Avg	95% CI
0.1	-79.016	9.85
1.0	-68.963	15.46
5.0	-86.701	12.33
10.0	-80.463	16.28
30.0	-80.827	6.54

ζ	Avg	95% CI
0.1	-291.268	10.91
1.0	-291.921	11.55
5.0	-254.954	28.81
10.0	-287.949	20.79
30.0	-268.919	17.84

Table 2: GP-UCB in search space dimensions 3, 10, and 20 after 500 evaluations respectively.

ω	Avg	95% CI
0.1	-1.30263	0.72
0.2	-1.49936	0.91
0.3	-0.73017	0.25
0.4	-1.19401	0.40
0.5	-0.48264	0.14

ω	Avg	95% CI
0.1	-0.27238	0.13
0.2	-0.25190	0.15
0.3	-0.32197	0.12
0.4	-0.21404	0.07
0.5	-0.27876	0.05

ω	Avg	95% CI
0.1	-1.04842	0.02
0.2	-1.08178	0.0.04
0.3	-1.17069	0.04
0.4	-1.26585	0.09
0.5	-1.56270	0.16

Table 3: VOO in search space dimensions 3, 10, and 20 after 500 evaluations.

C	Avg	95% CI
1e-07	-0.12154	0.05
1e-06	-0.09733	0.05
0.0001	-0.04930	0.02
0.001	-0.08428	0.06
0.01	-0.06187	0.03
0.1	-0.52441	0.23
1.0	-1.77516	0.24
5.0	-1.87757	0.61

C	Avg	95% CI
1e-07	-0.078	0.05
1e-06	-0.048	0.06
0.0001	-0.109	0.07
0.001	-0.104	0.12
0.01	-1.050	0.016
0.1	-27.4	18.94
1.0	-73.2	10.22
5.0	-75.1	11.12

C	Avg	95% CI
1e-07	-0.275	0.23
1e-06	-0.305	0.28
0.0001	-0.049	0.05
0.001	-0.219	0.24
0.01	-14.93	25.15
0.1	-132.93	11.25
1.0	-293.54	39.95
5.0	-290.45	64.75

Table 4: DOO in search space dimensions 3, 10, and 20 after 500 evaluations.

For SOO, we have

- 3D: -0.34484 ± 0.07
- 10D: -28.90801 ± 12.87
- 20D: -158.48764 ± 22.54

The Rastrigin function

ζ	Avg	95% CI	ζ	Avg	95% CI	ζ	Avg	95% CI
0.1	-7.81380	6.04	0.1	-32.08974	11.73	0.1	-157.87728	24.00
1.0	-2.84275	1.39	1.0	-78.83763	13.66	1.0	-191.15050	8.45
5.0	-5.40852	3.65	5.0	-47.53283	10.25	5.0	-179.43527	14.26
10.0	-5.44691	1.90	10.0	-54.52357	9.41	10.0	-167.24343	14.07
30.0	-11.11052	2.39	30.0	-36.11930	8.45	30.0	-182.79996	8.53

Table 5: GP-UCB in search space dimensions 3, 10, and 20 after 500, 1000, and 1000 evaluations respectively.

ω	Avg	95% CI	ω	Avg	95% CI	ω	Avg	95% CI
0.1	-7.18335	2.75	0.1	-49.65879	12.61	0.1	-115.16367	36.35
0.2	-5.27579	1.66	0.2	-43.48545	16.02	0.2	-108.62817	45.24
0.3	-5.37279	2.33	0.3	-31.80570	10.90	0.3	-96.66241	35.52
0.4	-4.17882	1.63	0.4	-35.25300	15.10	0.4	-84.90760	21.13
0.5	-5.27328	1.75	0.5	-41.73008	16.87	0.5	-101.41646	27.196

Table 6: VOO in search space dimensions 3, 10, and 20 after 500, 1000, and 1000 evaluations respectively.

C	Avg	95% CI	C	Avg	95% CI	C	Avg	95% CI
1e-07	-4.17902	1.63	1e-07	-63.95390	15.87	1e-07	-175.84	21.85
1e-06	-4.47732	1.99	1e-06	-45.37932	13.77	1e-06	-161.13	22.47
0.0001	-3.97983	1.96	0.0001	-71.11761	7.11	0.0001	-190.87	29.02
0.001	-6.21910	2.53	0.001	-58.74855	11.89	0.001	-166.08	26.97
0.01	-3.38286	1.00	0.01	-75.25954	15.39	0.01	-162.26	20.67
0.1	-3.26529	0.93	0.1	-56.03273	11.09	0.1	-167.94	20.49
1.0	-3.35029	1.11	1.0	-59.42687	13.51	1.0	-184.00	26.07
5.0	-4.17571	2.00	5.0	-85.72818	13.67	5.0	-212.27	15.99

Table 7: DOO in search space dimensions 3, 10 20 after 500, 1000, and 1000 evaluations respectively.

For SOO, we have

- 3D: -6.50919 ± 2.23
- 10D: -91.93641 ± 12.13
- 20D: -237.13848 ± 15.12

The Shekel function

ζ	Avg	95% CI
0.01	1.33357	0.46108
0.1	2.44774	0.46956
1.0	1.70730	0.62507
5.0	1.81943	0.92451
10.0	2.89227	0.95820
30.0	2.06799	0.78245

ζ	Avg	95% CI
0.01	0.00005	0.00001
0.1	0.00006	0.00001
1.0	0.00006	0.00001
5.0	0.00005	0.00001
10.0	0.00006	0.00001
30.0	0.00005	0.00000

ζ	Avg	95% CI
0.01	0.00002	0.00000
0.1	0.00001	0.00000
1.0	0.00002	0.00000
5.0	0.00001	0.00000
10.0	0.00002	0.00000
30.0	0.00002	0.00000

Table 8: GP-UCB in search space dimensions 3, 10 20 after 500, 1000, and 5000 evaluations respectively.

ω	Avg	95% CI
0.1	3.83946	0.43232
0.2	4.18810	0.47926
0.3	3.94591	0.33757
0.4	3.66445	0.63883
0.5	3.88531	0.56649

ω	Avg	95% CI
0.001	5.75182	1.61088
0.1	4.36805	1.90149
0.2	4.77551	1.94992
0.3	4.64517	1.48353
0.4	4.53959	1.88090
0.5	4.50249	1.96533

ω	Avg	95% CI
0.1	3.52871	0.99485
0.2	3.51387	0.97883
0.3	3.81839	0.99979
0.4	3.18553	0.96752
0.5	3.03880	0.78986

Table 9: VOO in search space dimensions 3, 10 20 after 500, 1000, and 5000 evaluations respectively.

C	Avg	95% CI
2e-16	4.11200	0.35866
1e-07	3.27224	0.60
1e-06	4.14476	0.42
0.0001	0.00611	0.01
0.001	0.00541	0.00
0.01	0.00309	0.00
0.1	0.00537	0.01
1.0	0.00346	0.00

C	Avg	95% CI
2e-16	1.21913	1.60
1e-07	0.00006	0.00
1e-06	0.00005	0.00
0.0001	0.00005	0.00
0.001	0.00005	0.00
0.01	0.00005	0.00

C	Avg	95% CI
2e-16	0.0996	0.02
1e-07	0.000	0.00
1-07	0.0000	0.00
1e-06	0.0000	0.00
0.0001	0.0000	0.00
0.001	0.0000	0.00
0.01	0.0000	0.00
0.1	0.0000	0.00
1.0	0.0000	0.00

Table 10: DOO in search space dimensions 3, 10 20 after 500, 1000, and 5000 evaluations respectively.

For SOO, we have

- 3D: 3.93956 ± 0.39
- 10D: 0.07724 ± 0.03
- 20D: 0.00029 ± 0.00

Robotics domains

We now describe the details of the robotics domains. It is particularly important to solve these problems with as few simulations as possible, because evaluating each potential action sample requires performing costly kinematic analysis and motion planning. In both domains, the reward function is discontinuous, with many flat, suboptimal regions; in addition, there are "dead ends," in which an action choice with high local reward that happens early in the plan may render the rest of the problem infeasible.

For both of the domains, the feasibility of any action sample requires checking kinematic solutions and collisions of the specified robot configurations, as well as determining the existence of a collision-free path. For checking the kinematics solution, we used IKFast in OpenRAVE [3]. For checking the existence of a collision free path, we used bidirectional Rapdily-exploring-Randomized-Trees (RRT) [7].

Unlike the standard problems in which continuous Monte Carlo planning algorithms have been applied in the past, such as inverted pendulum, our problems contain many infeasible actions. So, instead of executing an action based on the number of visits to the root node, we execute an action if we have 10 feasible actions at the root node.

For policy search algorithms, we used poses of objects as the state representation. Since our problems have the characteristic in which most of the actions are infeasible, we implement the decaying-epsilon-greedy exploration strategy where the epsilon decays with the number of *feasible trajectories*. We define a trajectory to be *feasible* if it contains at least one feasible action. Denote the number of feasible trajectories as n. Recall that we denote the action space as U. Then, the decaying-epsilon-greedy exploration policy is implemented as:

$$u \sim Unif(U) \text{ with probability } \epsilon^n$$
$$u \sim [\pi(s) + \mathcal{N}] \text{ with probability } 1 - \epsilon^n$$

where $\mathcal{N}$ denotes the Gaussian random noise. We used $\epsilon = 0.95$. The purpose of this particular exploration strategy is that initially, the randomly-initialized policies samples obviously-infeasible actions, such as placements outside the room. By sampling uniformly within the given bounded action space, it is much more likely to sample a feasible action.

For PPO, the hyperparameter is the clipping factor, and for DDPG, it is the soft-update rate. We test three different values, including the values suggested in the original papers, and report the avgerage and CI obtained by executing the algorithms with 20 different random seeds. We will denote the hyperparameters in each policy search algorithm as τ.

Object clearing domain In this domain, the order in which the robot should move the obstacles is given to it; the search tree is formed by selecting the parameters for "pick" and "place" actions in alternating layers. *Pick* actions have 6 dimensions: the base pose in bounded SE(2) and three grasp parameters. *Place* actions have 3 dimensions, specifying a base pose in bounded SE(2).

One primary difficulty that we could not mention in the main paper due to the space constraint is that, for RAND-DOOT, there are large parts of the kitchen region that constitute "dead-ends" in the sense that placing an object there will prevent future success of the plan. These are attractive because they involve moving the object only a short distance, so they have high immediate reward. VOOT is able to escape these local optima more quickly, as soon as it has found a better alternative even once.

The reward function for this domain is

$$r(s,a) = \begin{cases} 0 & \text{feasible pick} \\ \min\left(\frac{1}{g(o,o')}, r_{goal}\right) & \text{feasible place} \wedge \text{cleared} \\ \max(-g(o,o'), r_{min}) & \text{feasible place} \wedge \text{not cleared} \\ r_{min} & \text{otherwise (infeasible action)} \end{cases}$$

where r_{min} is set to -2, and r_{goal} is set to 2, and $g(o, o')$ is the distance between the object's original pose, o, to the new pose o'. For this domain, $H = 14$ and $\gamma = 0.9$. For VOOT and RAND-DOOT, we use: $N_r = 5, \kappa_r = 0.99$.

We now report the results. Each table contains the average rewards and 95% confidence interval with different N_{iter} values, obtained from 20 different random seeds.

UCT	Widening ratio	$N_{iter} = 500$	$N_{iter} = 1000$	$N_{iter} = 2000$
1.00	0.30	0.8943 ± 0.3121	1.85590 ± 0.3541	2.8989 ± 0.3911
0.10	0.80	2.3694 ± 0.3411	3.41537 ± 0.4763	4.2577 ± 0.4870
1.00	0.50	1.4985 ± 0.3506	2.38431 ± 0.3731	3.1803 ± 0.5387
0.01	0.80	2.1903 ± 0.3604	3.04347 ± 0.3888	4.0024 ± 0.3908
10.00	0.80	1.8651 ± 0.2501	2.43562 ± 0.3343	3.3646 ± 0.5629
1.00	0.80	2.0842 ± 0.3440	2.64119 ± 0.4745	3.5779 ± 0.5723

Table 11: PW-UCT in the object clearing domain

ω	$N_{iter} = 500$	$N_{iter} = 1000$	$N_{iter} = 2000$
0.200	2.9681 ± 0.2784	3.94391 ± 0.3630	4.2727 ± 0.3519
0.050	3.3022 ± 0.2192	4.22073 ± 0.3381	4.3571 ± 0.3218
0.010	3.1012 ± 0.2333	3.93188 ± 0.3741	4.2037 ± 0.3314
0.100	3.3949 ± 0.2322	4.50443 ± 0.2282	4.5607 ± 0.2341
0.300	2.7348 ± 0.3834	4.06344 ± 0.3847	4.2048 ± 0.3702
0.400	3.0348 ± 0.2064	4.33921 ± 0.2661	4.4998 ± 0.1638

Table 12: VOOT in the object clearing domain

C	$N_{iter} = 500$	$N_{iter} = 1000$	$N_{iter} = 2000$
1.000	2.5131 ± 0.1780	3.27900 ± 0.2335	4.2631 ± 0.2870
10.000	2.1686 ± 0.2778	3.07667 ± 0.2598	3.9274 ± 0.3307
2.000	2.4326 ± 0.1754	3.42675 ± 0.2763	4.2158 ± 0.3426
0.100	2.3843 ± 0.2565	3.33620 ± 0.4017	4.2245 ± 0.3737
100.000	2.3071 ± 0.1926	3.33670 ± 0.2414	4.2888 ± 0.2367
0.001	2.6039 ± 0.1639	3.68434 ± 0.2172	4.3723 ± 0.2260

Table 13: RAND-DOOT in the object clearing domain

τ	$N_{iter} = 500$	$N_{iter} = 1000$	$N_{iter} = 2000$
0.1	1.4493 ± 0.2007	1.6955 ± 0.1783	1.8673 ± 0.1501
0.2	1.2701 ± 0.1944	1.5575 ± 0.2030	1.7150 ± 0.2010
0.3	1.3700 ± 0.1911	1.5799 ± 0.1786	1.7274 ± 0.1451

Table 14: PPO in the object clearing domain

τ	N_{iter} = 500	N_{iter} = 1000	N_{iter} = 2000
1e-4	1.0756 ± 0.1558	1.1182 ± 0.1586	1.1824 ± 0.1449
1e-3	0.9434 ± 0.1156	1.1285 ± 0.1326	1.2025 ± 0.1350
1e-2	0.8830 ± 0.0952	0.9676 ± 0.1397	1.0680 ± 0.1405

Table 15: DDPG in the object clearing domain

Conveyor belt In this domain, the robot's objective is to receive 20 boxes with various sizes from a conveyor belt and pack them into rooms with narrow entrances; the primary challenge, which we could not mention in the paper, is that in this domain the first two boxes are too big to go through the door that leads to the bigger storage rooms, so it must place them in the first storage room, which is small, in such a way that there is still room for moving the rest of the objects into the bigger rooms.

In this domain, the order of objects is pre-specified and the grasp parameters already chosen, so there are only "place" actions. But because the placements are highly interdependent, we determine placements for groups of three objects at a time, so the actions space at each search node is nine-dimensional.

The reward function for this domain is

$$r(s,a) = \begin{cases} 1/g(c_0, c_{place}) & \text{feasible placement} \\ r_{min} & \text{action infeasible} \\ 2 & \text{20 boxes placed} \end{cases}$$

For this domain, $H = 7, \gamma = 0.99$. For VOOT and RAND-DOOT, we use: $N_r = 5, \kappa_r = 0.99$. We now report the results. Each table contains the average rewards and 95% confidence interval with different N_{iter}, obtained from 50 different random seeds.

UCT	Widening ratio	N_iter = 750	N_{iter} = 1500	N_{iter} = 3000
1.00	0.30	−1.1039 ± 0.3909	−0.16558 ± 0.5305	1.2532 ± 0.5186
1.00	0.50	−0.9328 ± 0.3631	0.21792 ± 0.3118	0.9886 ± 0.2304
0.10	0.60	−0.5564 ± 0.5046	1.25889 ± 0.5112	2.1764 ± 0.3191
0.10	0.80	0.2960 ± 0.5159	1.14080 ± 0.4847	2.0934 ± 0.3136
10.00	0.50	−0.9867 ± 0.3462	0.03704 ± 0.3034	0.4950 ± 0.2208
10.00	0.80	−0.5040 ± 0.3415	−0.08414 ± 0.3179	0.7124 ± 0.2571
1.00	0.80	−0.1770 ± 0.4301	0.46379 ± 0.3956	1.2432 ± 0.2646

Table 16: PW-UCT in the conveyor belt domain

ω	N_{iter} = 750	N_{iter} = 1500	N_{iter} = 3000
0.200	2.4777 ± 0.7311	3.34186 ± 0.5924	3.7182 ± 0.5413
0.100	2.5465 ± 0.7535	3.53981 ± 0.6219	3.9460 ± 0.5998
0.300	2.3714 ± 0.7434	3.49852 ± 0.6691	3.9361 ± 0.6198
0.400	2.0300 ± 0.6748	3.10781 ± 0.6206	3.6459 ± 0.5835

Table 17: VOOT in the conveyor belt domain

C	N_{iter} = 750	N_{iter} = 1500	N_{iter} = 3000
1.0	0.4493 ± 0.3763	1.04231 ± 0.2358	1.6230 ± 0.3254
10.0	0.3529 ± 0.4248	1.26372 ± 0.2328	1.7375 ± 0.2089
0.01	0.1233 ± 0.4138	0.93016 ± 0.2958	1.7338 ± 0.2383
0.1	0.2292 ± 0.4016	1.09811 ± 0.2021	1.6533 ± 0.2737

Table 18: RAND-DOOT in the conveyor belt domain

τ	$N_{iter} = 500$	$N_{iter} = 1000$	$N_{iter} = 2000$
0.1	-0.6229 ± 0.4685	-0.0417 ± 0.4033	0.4803 ± 0.2549
0.2	-0.4399 ± 0.5888	-0.0831 ± 0.5223	0.3414 ± 0.3354
0.3	-0.7038 ± 0.4711	-0.1461 ± 0.4740	0.5266 ± 0.3788

Table 19: PPO in the conveyor belt domain

τ	$N_{iter} = 500$	$N_{iter} = 1000$	$N_{iter} = 2000$
1e-4	-0.4775 ± 0.5476	-0.0133 ± 0.5171	0.7199 ± 0.2863
1e-3	-0.8331 ± 0.4307	-0.0168 ± 0.4008	0.4032 ± 0.3090
1e-2	-0.8281 ± 0.4141	-0.0164 ± 0.4404	0.3260 ± 0.3375

Table 20: DDPG in the conveyor belt domain

References

[1] D. Auger, A. Couëtoux, and Olivier Teytaud. Continuous Upper Confidence Trees with polynomial exploration - consistency. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013.

[2] Bille C Carlson. Some inequalities for hypergeometric functions. *Proceedings of the American Mathematical Society*, 17(1):32–39, 1966.

[3] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, CMU Robotics Institute, August 2010.

[4] GPy. GPy: A Gaussian Process framework in Python. `http://github.com/SheffieldML/GPy`, since 2012.

[5] Anatoly A Kilbas. *H-transforms: Theory and Applications*. CRC Press, 2004.

[6] Kyeong-Hun Kim and Sungbin Lim. Asymptotic behaviors of fundamental solution and its derivatives to fractional diffusion-wave equations. *J. Korean Math. Soc*, 53(4):929–967, 2016.

[7] J.J Kuffner and S.M LaValle. RRT-connect: An efficient approach to single-query path planning. In *International Conference on Robotics and Automation*, 2000.

[8] R. Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. *Advances in Neural Information Processing Systems*, 2011.

[9] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.