Journal of Machine Learning Research 25 (2024) 1-40 Submitted 2/22; Revised 10/23; Published 3/24

Materials Discovery using Max K-Armed Bandit

Nobuaki Kikkawa KIKKAWA@MOSK.TYTLABS.CO.JP
Toyota Central R&D Labs., Inc.
41-1, Yokomichi, Nagakute, Aichi 480-1192, Japan

Hiroshi Ohno OONO-H@MOSK.TYTLABS.CO.JP
Toyota Central R&D Labs., Inc.
41-1, Yokomichi, Nagakute, Aichi 480-1192, Japan

Editor: Aurelien Garivier

Abstract

Search algorithms for bandit problems are applicable in materials discovery. However, objectives of the conventional bandit problem are different from those of materials discovery. The conventional bandit problem aims to maximize the total rewards, whereas materials discovery aims to achieve breakthroughs in material properties. The max K-armed bandit (MKB) problem, which aims to acquire the single best reward, matches with the discovery tasks better than the conventional bandit. However, typical MKB algorithms are not directly applicable to materials discovery due to some difficulties. The typical algorithms have many hyperparameters and some difficulty in the directly implementation for the materials discovery. Thus, we propose a new MKB algorithm using an upper confidence bound of expected improvement of the best reward. This approach is guaranteed to be asymptotic to greedy oracles, which does not depend on the time horizon. In addition, compared with other MKB algorithms, the proposed algorithm has only one hyperparameter, which is advantageous in materials discovery. We applied the proposed algorithm to synthetic problems and molecular-design demonstrations using a Monte Carlo tree search. According to the results, the proposed algorithm stably outperformed other bandit algorithms in the late stage of the search process, unless the optimal arm coincides in the MKB and conventional bandit settings.

Keywords: Max K-armed bandit problem, Confidence bounds, Monte Carlo tree search, Molecular design, Greedy oracle

1. Introduction

Materials discovery integrated with machine learning is a field with immense growth potential. Material property predictions using regression and clustering methods (Liu et al., 2017; Meredig et al., 2018; Butler et al., 2018; Ramprasad et al., 2017; Pilania et al., 2013) are recognized as a beneficial approach in the development workplace. Materials discovery using deep learning (Agrawal and Choudhary, 2019; Jha et al., 2018), transfer learning (Jha et al., 2019; Yamada et al., 2019), and generative models (Sanchez-Lengeling et al., 2017; Sanchez-Lengeling and Aspuru-Guzik, 2018) is actively under investigation in advanced researches. Autonomous searches based on Bayesian optimization (Ueno et al., 2016; Kusne et al., 2020), Monte Carlo tree search (MCTS) (M. Dieb et al., 2017; Yang et al., 2017; Ju et al., 2018; Segler et al., 2018; Kiyohara and Mizoguchi, 2018; M. Dieb et al., 2018; Ka-

jita et al., 2020; Kikkawa et al., 2020; Patra et al., 2020), and reinforcement learning (RL) (Sanchez-Lengeling et al., 2017; Popova et al., 2018; Olivecrona et al., 2017) have also been investigated to accelerate materials discovery. Active learning approaches (Kusne et al., 2020; Del Rosario et al., 2020) and the effective use of failed experiments (Raccuglia et al., 2016) are also important for overcoming the limitations of data generation in materials science.

Finding novel materials with record-breaking properties of interest is one of the goals of materials discovery. However, the guiding principles of MCTS and RL seem to differ from the goal of materials discovery because these approaches mainly focus on maximizing the total reward (Auer et al., 2002; Kocsis and Szepesvári, 2006; Browne et al., 2012; Sutton and Barto, 2018) rather than discovering a record-breaking material property. Therefore, these approaches tend to avoid selections with high failure rates even though those could lead to a few great breakthroughs. Because failure is often a prerequisite for success, these approaches are not always optimal for achieving a significant discovery.

The max K-armed bandit (MKB) problem (Cicirello and Smith, 2005), also called the extreme bandit (Carpentier and Valko, 2014) or the max bandit (David and Shimkin, 2016), is a promising problem setting for materials discovery. In the MKB problem, a player aims to maximize the single best reward from a slot with K arms instead of the total reward in the conventional bandit problem (Lai and Robbins, 1985). Owing to these modifications, the algorithms for the MKB problem can explore the adventurous arm rather than the stable arm.

Several algorithms have been proposed for the MKB problem (Carpentier and Valko, 2014; David and Shimkin, 2016; Streeter and Smith, 2006b; Achab et al., 2017; Streeter and Smith, 2006a; Baudry et al., 2022; Bhatt et al., 2022). However, their practical applications in materials discovery are limited. Some of them consider the time horizon T as a hyperparameter, even though their applications for MCTS are associated with many drawbacks. Other methods involve many hyperparameters depending on unknown reward distributions. This requires a time-consuming parameter tuning, which is extremely costly for materials discovery. To overcome these difficulties, we propose an MKB algorithm with one hyperparameter that employs an upper confidence bound (UCB) of the expected improvement (EI) of the maximum reward as the selection index of the arm. We apply this algorithm to synthetic problems and demonstrations of materials discovery using MCTS.

The primary contributions of this study are as follows:

1. We propose MKB algorithms by introducing the UCB of EI of the best reward[1], which has only one hyperparameter to control the balance between exploration and exploitation.

2. We demonstrate that the MCTS approach based on the MKB algorithm is effective for materials discovery than other MCTS algorithms based on the conventional bandit algorithm.

3. We prove that asymptotically optimal MKB algorithms can be generated using the UCB of EI of the best reward.

1. An algorithm is based on a tentative value of the UCB of the EI.

4. We propose a time-independent oracle named *Kikkawa's greedy oracle*. This oracle makes it possible to discuss the MKB problem in the almost same manner as the conventional bandit problem.

5. To the best of our knowledge, this is the first study to actually apply the MKB algorithm to materials discovery.

The remainder of this paper is structured as follows. Section 2 describes the related work of the MKB problem, materials discovery using MCTS, and related algorithms. After that, we define some terms and representations in Section 3. In Section 4, we describe the idea to create an MKB algorithm. In the following Section 5, the derivation of the proposed algorithm is presented. Section 6 demonstrates the experiments conducted for comparing the proposed algorithm with other bandit algorithms. In Section 7, we discuss the subtleties of the MKB problem and theoretical aspect of the role of the UCB of EI for the MKB problem. Finally, Section 8 presents our conclusions and discussions on the future outlook of the proposed algorithm.

2. Related Work

In this section, we first describe the related work of the MKB problem, materials discovery using MCTS, and related algorithms.

2.1 Max K-armed bandit problem

The MKB problem is expressed as a policy-decision problem that maximizes the single best reward $\max_{t\in[T]} r_{k(t)}(t)$, where $[T] \coloneqq \{1, 2, \ldots, T\}$, $r_{k(t)}$ is the reward from the k-th arm at time t with a time-independent distribution $f_k(r)$, and $k(t)$ is the selected arm index at time t determined based on the policy to be tuned (Cicirello and Smith, 2005). This is a simple variant of the conventional bandit problem, which aims to maximize the total reward $\sum_{t\in[T]} r_{k(t)}(t)$ (Lai and Robbins, 1985).

The MKB problem was first proposed by Cicirello and Smith (2005), who derived the optimal allocation order for this problem with the Gumbel-type reward distribution. The following year, Streeter and Smith (2006a) proposed an asymptotically optimal algorithm using the explore-then-commit (ETC) approach. They also proposed a UCB algorithm for the MKB problem, called *ThresholdAscent*, in the same year (Streeter and Smith, 2006b). This algorithm used a UCB of $\mathbb{E}[\mathbb{1}[r_k > r^{s\text{-th}}]]$ as the selection index, where $r^{s\text{-th}}$ was the s-th maximum of observed rewards.

The next stream of the algorithm development for the MKB problem was undertaken by Carpentier and Valko (2014). They estimated a finite-time upper bound of $\mathbb{E}[r^{\max}]$ assuming the reward distribution as the second-order Pareto distribution and proposed *ExtremeHunter* algorithm based on it. Achab et al. (2017) proposed the ETC version of *ExtremeHunter*, and they also proposed a simple algorithm, denoted *RobustUCBMax* in this paper. The *RobustUCBMax* used a robust UCB (Bubeck et al., 2013) of $\mathbb{E}[r_k \mathbb{1}[r_k > u]]$, where u was a threshold parameter. A probably approximately correct (PAC) approach for the MKB problem called *Max-CB* was discussed theoretically by David and Shimkin (2016).

Recently, distribution-free (DF) approaches were used to create new MKB algorithms. Bhatt et al. (2022) proposed *Max-Median* algorithm. In their algorithm, the order statistic, which estimates the median of maxima in conceivable subsets of observed rewards, was adopted as the selection index. Baudry et al. (2022) also employed the quantile of maxima (QoMax) as an extension of the median of maxima, and developed ETC algorithm and subsampling dueling algorithm (SDA) using QoMax, which was directly calculated from the subsets of observed rewards.

Here, we summarize the features of the MKB algorithms in Table 1. Additionally, we show the main target of these algorithms in this table. The MKB algorithms have not been applied for materials discovery in the previous studies, although it was discussed by David and Shimkin (2016).

Table 1: MKB algorithms. The term "Anytime" refers to algorithms that do not employ T as a hyperparameter.

Algorithm	Approach	Anytime	# of parameters	Target
MaxSearch (this work)	UCB	yes	1	synthetic problem, materials discovery
asymptotically algorithm	ETC	no	2	-
ThresholdAscent	UCB	no	2	scheduling
ExtremeHunter	finite-time upper bound	no	5	synthetic problems, traffic analysis
ExtremeETC	ETC	no	5	synthetic problem
RobustUCBMax	UCB	yes	3	synthetic problem
Max-CB	PAC	yes	2	-
MaxMedian	DF	yes	2	synthetic problem
QoMax-ETC	DF-ETC	no	3	synthetic problem
QoMax-SDA	DF-SDA	yes	3	synthetic problem

Some researchers also contributed theoretically. Cicirello and Smith (2005) stated that in the case of the MKB problem with the Gumbel-type reward distributions, the optimal algorithm should sample the observed best arm at a rate increasing double exponentially relative to the other arms. Carpentier and Valko (2014) introduced an expected regret, called *extreme regret*, for the MKB problem. They also proposed an algorithm where the regret had $o(\mathbb{E}\left[\max_{t\in[T]} r_k(t)\right])$. Although these theoretical developments were traced to the analogies of the conventional bandit problem, Nishihara et al. (2016) proved that no policy is guaranteed to asymptotically approach the oracle used by Carpentier and Valko (2014) in some settings. Nishihara et al. (2016) also pointed out some other subtleties on the MKB problem and proposed an oracle using EI although they does not analyze it much.

2.2 Materials discovery using Monte Carlo tree search

There are several studies relating to materials discovery using MCTS (M. Dieb et al., 2017; Yang et al., 2017; Ju et al., 2018; Segler et al., 2018; Kiyohara and Mizoguchi, 2018; M. Dieb et al., 2018; Kajita et al., 2020; Kikkawa et al., 2020; Patra et al., 2020). M. Dieb et al.

(2017), in the pioneering work, compiled Si-Ge interfacial conformations into binaries and optimized them to maximize the thermal conductance using MCTS. Ju et al. (2018) also optimized the interface roughness by ternary embedding. The optimizations of the grain boundary (Kiyohara and Mizoguchi, 2018), doping (M. Dieb et al., 2018), and chemical syntheses (Segler et al., 2018; Patra et al., 2020) have also been investigated.

Yang et al. (2017) applied MCTS to the optimization of chemical structures. They introduced a search tree in which nodes correspond to the simplified molecular-input line-entry system (SMILES) characters (Weininger, 1988), e.g., "C", "O", "(", and ")". Because the SMILES grammar can express most of molecules, the chemical-structure optimization is regarded as a string optimization in this approach. They showed that the MCTS approach outperformed other approaches in the SMILES search.

The MCTS approach using SMILES was employed in subsequent studies. Kajita et al. (2020) introduced fragments of SMILES, such as "CC" and "CO" to restrict the search space of chemical structures. In their study, they attempted 5,500 evaluations 10 times using molecular dynamics (MD) simulations in a search run. They also confirmed the properties of the molecules with high rewards by synthetic experiments. Kikkawa et al. (2020) improved the flexibility of the restriction by introducing rule-based grammar into the search tree using a maze game. They also evaluated several thousand molecules in a search run using MD simulations.

2.3 Other algorithms

The application of the single-player MCTS (Schadd et al., 2008) for materials discovery has also been considered. In this approach, a variance-dependent term is empirically added to the selection index of the UCB. Herein, we denote the bandit algorithm using this modified index *spUCB*.

We note that the best-arm identification, such as the *UCBE* algorithm (Audibert et al., 2010), is different from the MKB algorithm. The best-arm identification aims to find the arm with the maximum "expectation" reward not the "single" maximum through a search run. The algorithms based on the best-arm identification barely select arms with a low expectation reward even if the arm affords a high reward at low rates.

3. Definitions

The definitions used in this section through Section 5 are listed as follows:

Definition 1 (Bandit problem) *The K-armed bandit problem, or simply bandit problem, is a problem to maximize (minimize) some objective*

$$G\left[\{k(t)\}_{t\in[T]}; \{r_k(t)\}_{k\in[K],t\in[T]}\right] \tag{1}$$

in a selection game with K arms during time horizon T, where $k(t), t \in [T]$ is a player's selection which should be optimized. The arm $k \in [K]$ returns a reward $r_k(t), t \in [T]$ at time t, following unknown time-independent reward distribution $f_k(r)$. A player also does not know T in the "anytime" setting. We usually omit the dependency of G on $k(t)$ and $r_k(t), k \in [K]$, and $t \in [T]$.

Definition 2 (Conventional bandit problem) *The conventional bandit problem is a bandit problem to maximize the total reward*

$$G^{sum}\left[\{k(t)\}_{t\in[T]}\right] \coloneqq \sum_{t\in[T]} r_{k(t)}(t). \tag{2}$$

Definition 3 (MKB problem) *The MKB problem is a bandit problem to maximize the single maximum reward*

$$G^{max}\left[\{k(t)\}_{t\in[T]}\right] \coloneqq \max_{t\in[T]} r_{k(t)}(t). \tag{3}$$

Definition 4 (EI) *In the bandit problem, the EI of arm k at time $\tau \leq T$ is defined as*

$$EI\left[k,\tau;G\right] \coloneqq \mathbb{E}_{f_k}\left[G\left[\{\tilde{k}(t)\}_{t\in[\tau]}\right]\right] - \mathbb{E}_{f_k}\left[G\left[\{k(t)\}_{t\in[\tau-1]}\right]\right], \tag{4}$$

where $\tilde{k}(t) = k(t)$ when $t \in [\tau-1]$ and $\tilde{k}(t) = k$ when $t = \tau$.

Definition 5 (Complementary error function)

$$\mathrm{erfc}(x) \coloneqq \frac{2}{\pi}\int_x^\infty \exp(-x^2)dx. \tag{5}$$

Definition 6 (Integral of $\mathrm{erfc}(x)$ (Olver et al., 2010))

$$\mathrm{ierfc}(x) \coloneqq \int_x^\infty \mathrm{erfc}(x)dx = \frac{1}{\sqrt{\pi}}\exp(-x^2) - x\ \mathrm{erfc}(x). \tag{6}$$

Definition 7 (Sub-gaussian) *A distribution $f(r)$ is called a sub-gaussian distribution when $\exists m \in \mathbb{R}$ and $\exists s \in \mathbb{R}^+$, such that*

$$\mathbb{P}_f[|r| < u] \leq U(u;m,s^2), \tag{7}$$

where

$$U(r;m,s^2) \coloneqq 2\exp\left[-\frac{(r-m)^2}{2s^2}\right]. \tag{8}$$

The m and s^2 are called the mean and variance proxies, respectively.

Definition 8

$$I(r;m,s^2) \coloneqq \int_r^\infty U(u;m,s^2)du = \sqrt{2\pi s^2}\ \mathrm{erfc}\left[\frac{r-m}{\sqrt{2s^2}}\right]. \tag{9}$$

Definition 9 (Tentative upper bound) *The symbol $\lessapprox$ means that the right value is a tentative value of the upper bound of the left value.*

Definition 10 (Sub-exponential) *A distribution $g(x)$ is called a sub-exponential when $\exists b \geq 0$, such that*

$$\mathbb{P}_{g(x)}\{x \geq u\} \leq 2\exp\left(-\frac{u}{b}\right). \tag{10}$$

4. Our Concept

Our main claim in this article is the effectiveness of Algorithm 1 which uses a UCB of EI of the single best reward as the selection index. We first show the reasonability of the use of a UCB of EI for the bandit algorithm by taking the conventional UCB (Auer et al., 2002; Bubeck et al., 2013) as an example. This example is intuitively clear although it is not rigorous. We provide a more theoretical discussion in Section 7. We also show in Lemmas 13 and 14 that the EI of the single best reward can be calculated from a survival function of $f_k(r)$. The substantial value is estimated in Theorem 16 in the next section.

In the conventional bandit problem, the EI becomes the expected reward as shown in the following lemma.

Lemma 11 (EI of conventional bandit problem) *In the conventional bandit problem,*

$$EI\left[k, \tau; G^{sum}\right] = \mathbb{E}_{f_k}[r]. \tag{11}$$

Based on this lemma, we can consider that the conventional UCB algorithm (Auer et al., 2002) uses a UCB of $EI[k, \tau; G^{\mathrm{sum}}]$ as a selection index. This relation of the conventional UCB and EI implies that the same approach is valid in the MKB problem as follows.

Theorem 12 (MaxSearch strategy) *Let $z_k \coloneqq z(k,\, \mathcal{R}(\tau-1);\, G^{max})$ be a UCB of $EI[k, \tau; G^{max}]$, where $\mathcal{R}(\tau) \coloneqq \left\{\{k(t), r_{k(t)}(t)\}\right\}_{t\in\tau}$ is the set of the pairs of the selected arm ids and rewards previously played and obtained. The strategy which selects*

$$\underset{k\in[K]}{\operatorname{argmax}}\, z(k, \mathcal{R}(\tau-1); G^{max}) \tag{12}$$

in each selection (Algorithm 1) is an asymptotically optimal approach in the MKB bandit problem.

Proof See Section 7. ■

Algorithm 1 MaxSearch

Input: number of arms K, current time τ, and previous records $\mathcal{R}(\tau-1)$.
Output: selected arm index $\hat{k}$.

1: **for each** $k \in [K]$ **do**
2: calculate $z_k = z(k, \mathcal{R}(\tau-1); G^{\max})$
3: **end for**
4: $\hat{k} \leftarrow \underset{k\in[K]}{\operatorname{argmax}}\, z_k$
5: **return** $\hat{k}$

Theorem 12 does not state the explicit form of $z(k, \mathcal{R}(\tau-1); G^{\max})$. Therefore, we should estimate it to use Algorithm 1. The lemmas are footholds for the estimation.

Lemma 13 (EI of MKB problem) *In the MKB problem, let $r^{max} \coloneqq \max_{t\in[\tau-1]} r_{k(t)}(t)$ be given. Then,*

$$EI[k, \tau; G^{max}] = \mathbb{E}_{f_k}\left[\max\{r_{k(t)}(t), r^{max}\}\right] - r^{max}. \tag{13}$$

Lemma 14 (EI and survival function) *Let r be an independent identical distributed (i.i.d.) random variable following $f(r)$ and r_0 be given. Then,*

$$\mathbb{E}_f\left[\max\left\{r, r_0\right\}\right] - r_0 = \int_{r_0}^{\infty} S(u)du, \tag{14}$$

where

$$S(r) \coloneqq \int_{r}^{\infty} f(u)du \tag{15}$$

is the survival function of $f(r)$.

Proof

$$\begin{aligned}
\mathbb{E}_f\left[\max\left\{r, r_0\right\}\right] &= \int_{-\infty}^{r_0} r_0 f(r)dr + \int_{r_0}^{\infty} r f(r)dr \\
&= \int_{-\infty}^{r_0} r_0 f(r)dr + \int_{r_0}^{\infty}\int_{0}^{r} du f(r)dr \\
&= \int_{-\infty}^{r_0} r_0 f(r)dr + \int_{0}^{r_0} du \int_{r_0}^{\infty} f(r)dr + \int_{r_0}^{\infty} du \int_{u}^{\infty} f(r)dr \\
&= r_0 + \int_{r_0}^{\infty} S(u)du.
\end{aligned} \tag{16}$$

Here, in switching the order of the integration, we used

$$r_0 \leq r \cap 0 \leq u \leq r \Leftrightarrow (0 \leq u \leq r_0 \cap r_0 \leq r) \cup (r_0 \leq u \cap u \leq r). \tag{17}$$

■

Lemma 13 assumes $r^{\max}$ given. It is no problem in the implementation because $r^{\max}$ can be recorded on a $O(1)$ memory. Lemma 14 says that the EI in Lemma 13 can be calculated from the survival function of the reward distribution. Because of this, the remained work to obtain the selection index is the estimation of a substantial form of a UCB of $\int_{r^{\max}}^{\infty} S_k(r)dr$ with some assumption for the reward distribution.

5. Estimation for Selection Index

In this section, we derive two concrete forms of the UCB of EI. One is derived strictly under Gaussian reward settings, and the other is derived approximately under sub-gaussian reward settings. As will be shown in the next section, the former appears to better match the tasks of materials discovery. However, the assumption for the reward distribution in the latter approach is looser than that in the former. Because the material properties of different materials groups are sometimes bounded do not lie on a continuum, the latter might be preferable in some cases of practical materials discovery.

5.1 Selection Index under Gaussian Reward Settings

Under Gaussian reward settings, the reward distributions are written as

$$f_k(r) = \mathcal{N}(r; \mu_k, \sigma_k^2), \tag{18}$$

where $\mathcal{N}(r;\mu,\sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. In this case, the confidence interval of $\int_r^\infty S_k(u)du$ is given as follows:

Proposition 15 (Confidence interval of $\int_r^\infty S(u)du$) *Let $f(r)$ be a Gaussian distribution with mean μ and variance σ^2. Let $S(r)$ be a survival function of $f(r)$ and let $r_i, i \in [n]$ be i.i.d. Gaussian random variables following $f(r)$. We then have,*

$$\sqrt{\frac{\hat{\sigma}_-^2}{2}}\ \mathrm{ierfc}\left(\frac{r-\hat{\mu}_-}{\sqrt{2\hat{\sigma}_-^2}}\right) \leq \int_r^\infty S(u)du \leq \sqrt{\frac{\hat{\sigma}_+^2}{2}}\ \mathrm{ierfc}\left(\frac{r-\hat{\mu}_+}{\sqrt{2\hat{\sigma}_+^2}}\right) \tag{19}$$

with confidential level $1-\alpha$, where

$$\hat{\mu}_\pm \coloneqq \bar{\mu} \pm t_{n-1,1-\frac{\alpha}{2}}\frac{\bar{\sigma}}{\sqrt{n}},\quad \hat{\sigma}_\pm^2 \coloneqq \frac{(n-1)\bar{\sigma}^2}{\chi^2_{n-1,\frac{1}{2}\mp\frac{1-\alpha}{2}}}, \tag{20}$$

$$\bar{\mu} \coloneqq \frac{1}{n}\sum_{i=1}^n r_i,\quad \bar{\sigma}^2 \coloneqq \frac{1}{n-1}\sum_{i=1}^n (r_i-\bar{\mu})^2. \tag{21}$$

Here, $t_{n,p}$ and $\chi^2_{n,p}$ are the p-quantiles of t and χ^2 distributions, respectively.

Proof The function

$$\int_r^\infty S(u)du = \sqrt{\frac{\sigma^2}{2}}\ \mathrm{ierfc}\left(\frac{r-\mu}{\sqrt{2\sigma^2}}\right) \tag{22}$$

is a monotonically increasing function of μ and σ^2. Therefore, its confidence interval can be simply obtained by substituting the lower and upper confidence bounds $\hat{\mu}_- \leq \mu \leq \hat{\mu}_+$ and $\hat{\sigma}_-^2 \leq \sigma^2 \leq \hat{\sigma}_+^2$ (Pishro-Nik, 2014) into the function. ■

From this proposition and Theorem 12, the selection index is obtained as

$$z_k = \sqrt{\frac{\hat{\sigma}_k^2}{2}}\ \mathrm{ierfc}\left(\frac{r-\hat{\mu}_k}{\sqrt{2\hat{\sigma}_k^2}}\right), \tag{23}$$

$$\hat{\mu}_k = \bar{\mu}_k + t_{n_k-1,1-\frac{\alpha}{2}}\frac{\bar{\sigma}_k}{\sqrt{n_k}},\quad \hat{\sigma}_k^2 = \frac{(n_k-1)\bar{\sigma}_k^2}{\chi^2_{n_k-1,\frac{\alpha}{2}}}, \tag{24}$$

$$\bar{\mu}_k = \frac{1}{n_k}\sum_{\substack{k(t)=k,\\ t\in[\tau-1]}} r_{k(t)}(t),\quad \bar{\sigma}_k^2 = \frac{1}{n_k-1}\sum_{\substack{k(t)=k,\\ t\in[\tau-1]}} (r_{k(t)}(t)-\bar{\mu}_k)^2, \tag{25}$$

where n_k is the number that the arm k is selected previously. The summations in the calculation of $\bar{\mu}_k$ and $\bar{\sigma}_k^2$ are also performed only when the arm k is selected. The parameter α, which determines the balance between exploration and exploitation, is determined as

$$\alpha = \nu^{-c^2}, \text{where } \nu = \sum_{k\in[K]} n_k. \tag{26}$$

This value, usually called the allocation order, is consistent with the optimal order of the conventional bandit problem (Auer et al., 2002). However, it is inconsistent with the double exponential order, which is optimal in the MKB with Gumbel-type reward distributions (Cicirello and Smith, 2005). Considering the uncertainty of the MKB problem discussed in Section 7, the optimal allocation order will be explored in future work. Equation (23) is numerically calculated using Algorithm 2. In the present experiments, we set $c = 1$.

Algorithm 2 Selection index under Gaussian settings

Input: total number of selections previously performed ν, number of times the target arm is selected n, sum of the rewards obtained from the target arm R, sum of the square rewards obtained from the target arm Q, the maximum reward obtained until the current time $r^{\max}$, and the hyperparameter c.

Output: selection index z.

if $n > 1$ **then**
  $\bar{\mu} \leftarrow R/n$
  $\bar{\sigma}^2 \leftarrow (Q - n\bar{\mu}^2)/(n-1)$
  $\alpha \leftarrow \nu^{-c^2}$
  $\hat{\mu} \leftarrow \bar{\mu} + t_{n-1,1-\alpha/2}\sqrt{\bar{\sigma}^2/n}$
  $\hat{\sigma}^2 \leftarrow (n-1)\bar{\sigma}^2/\chi^2_{n-1,\alpha/2}$
  $z \leftarrow \sqrt{\hat{\sigma}^2/2}\ \mathrm{ierfc}\left[(r^{\max} - \hat{\mu})/\sqrt{2\hat{\sigma}^2}\right]$
else
  $z \leftarrow \infty$
end if
return z

5.2 Selection Index under Sub-gaussian Reward Settings

Employing the sub-gaussian assumption in the reward distributions, we have

$$\int_r^\infty S_k(u)du \leq I(r; m_k, s_k^2). \tag{27}$$

Therefore, we can use a UCB of $I(r^{\max}; m, s^2)$ as the selection index . We could only estimate a tentative value of this UCB as follows:

Proposition 16 (UCB of $I(r; m, s^2)$) *Let $f(r)$ be a sub-gaussian distribution with mean proxy m and variance proxy s^2. Let $S(r)$ be a survival function of $f(r)$. Let $r_i, i \in [n]$ be i.i.d. sub-gaussian random variables following $f(r)$. Then,*

$$I(r; m, s^2) \lessapprox I(r; \hat{m}, \hat{s}^2) \tag{28}$$

with the confidential level $0 < 1 - \alpha < 1 - \exp[-(\sqrt{2} - \sqrt{2 - \ln 2})^2 n]$, or almost equivalently $n > -13.613 \ln \alpha > 0$, where

$$\hat{m} \coloneqq \frac{1}{n}\sum_{i=1}^n r_i, \tag{29}$$

$$\hat{s}^2 \coloneqq \frac{1}{2[\ln 2 - \gamma(\alpha, n)]} \left(\frac{1}{n} \sum_{i=1}^{n} r_i^2 - \hat{m}^2 \right), \tag{30}$$

and

$$\gamma(\alpha, n) \coloneqq \frac{\ln \alpha}{n} + 2\sqrt{\frac{-2 \ln \alpha}{n}}. \tag{31}$$

This proposition is weak because it states only a tentative value. However, Algorithm 1 with the selection index calculated from Algorithm 3, which is based on Proposition 16, performed well as shown in Section 6.

Algorithm 3 Selection index under sub-gaussian settings

Input: total number of selections previously performed ν, number of times the target arm is selected n, sum of the rewards obtained from the target arm R, sum of the square rewards obtained from the target arm Q, the maximum reward obtained until the current time $r^{\max}$, and the hyperparameter c.

Output: selection index z.

if $n = 0$ or $\nu < 2$ **then**
  $z \leftarrow \infty$
else
  $\beta \leftarrow c\sqrt{(\ln \nu)/n}$
  $\gamma \leftarrow -\beta^2 + 2\sqrt{2}\beta$
  if $\gamma > \ln 2$ **then**
    $z \leftarrow \infty$
  else
    $\hat{m} \leftarrow R/n$
    $\hat{s}^2 \leftarrow (Q/n - \hat{m}^2)/[2(\ln 2 - \gamma)]$
    $z \leftarrow \sqrt{2\pi \hat{s}^2}\ \mathrm{erfc}\left[(r^{\max} - \hat{m})/\sqrt{2\hat{s}^2}\right]$
  end if
end if
return z

The reason for the weakness of Proposition 16 is that the sub-gaussian assumption lacks an upper bound of s^2(Lemma 18). We alternatively use a lower bound of s^2 in this lemma to derive Proposition 16. The term "tentative" represents the theoretical inauthenticity of this treatment. This alternative is valid only if the lower bound captures the characteristics of the tail distributions, but the assumption appears to be reasonable from the results in Section 6. The derivation of Proposition 16 is based on Bernstein's inequality in Proposition 20 because the lower bound of s^2 contains the expected square value of the reward. The details are given in the following subsections.

5.2.1 Sub-Gaussian assumption

Using the sub-gaussian assumption, a UCB of $I(r; m, s^2)$ is given by the following conceptual lemma.

Lemma 17 (UCB of $I(r; m, s^2)$) *Consider n samples $\{r_i\}, i \in [n]$ taken from the sub-gaussian distribution $f(r)$ with mean proxy m and variance proxy s^2. Let $\hat{m}$ and $\hat{s}^2$ be UCBs of the mean and variance proxies with the confidence level $0 < 1-\alpha < 1$, respectively. Then,*

$$I(r; m, s^2) \leq I(r; \hat{m}, \hat{s}^2) \tag{32}$$

with the confidence level $1-\alpha$, when $r \geq m$.

Proof With the confidence level $1-\alpha$, $m \leq \hat{m}$ and $s^2 \leq \hat{s}^2$. $I(m; m, s^2) \leq I(m; \hat{m}, \hat{s}^2)$. $I(r; m, s^2)$ is monotonically decreasing for r and increasing for m and s^2. Then, $I(r; m, s^2) \leq I(r; \hat{m}, \hat{s}^2)$ when $r \geq m$. ■

In this lemma, UCBs denoted as $\hat{m}$ and $\hat{s}^2$ are virtual. Thus, we represent it using the term "conceptual". Unfortunately, as the following lemma indicates, even the upper bound of s^2 cannot be determined under the sub-gaussian assumption.

Lemma 18 (Bounds on s^2) *Let $f(r)$ is a sub-gaussian with mean proxy m and variance proxy s^2. Then,*

$$2s^2 \geq \frac{\mathbb{E}_f[(r-m)^2]}{\ln 2}. \tag{33}$$

There are no upper bounds on s^2.

Proof Lower bound: As an equivalent condition to the sub-gaussian on Definition 7, the following Orlicz condition is established (Vershynin, 2018).

$$\mathbb{E}_f\left[\exp\left[\frac{(r-m)^2}{2s^2}\right]\right] \leq 2. \tag{34}$$

Applying Jensen's inequality to this condition, we obtain $\exp\left[\mathbb{E}_f[(r-m)^2]/(2s^2)\right] \leq 2$. Then, $2s^2 \geq \mathbb{E}_f[(r-m)^2]/\ln 2$.

No upper bound: From the sub-gaussian definition, $f(r) \leq U(r; m, s^2)$. Then, $f(r) \leq U(r; m, \hat{s}^2)$, where $\hat{s}^2 > s^2$. Therefore, $f(r)$ can be considered as a sub-gaussian with variance proxy $\hat{s}^2 > s^2$. It means no upper bound of s^2. ■

Because of this lemma, we gave up deriving a rigor UCB of s^2. Alternatively, we decided to use a UCB of the lower bound $\mathbb{E}_f[(r-m)^2]/\ln 2$ as a tentative UCB.

5.2.2 Derivation for Theorem 16

To estimate a UCB of $\mathbb{E}_f[(r-m)^2]/\ln 2$, we first indicate the sub-exponential property of $(r-m)^2$ in Lemma 19. A UCB of the expected value of the sub-exponential distribution is known to be estimated from Bernstein's inequality (Vershynin, 2018). Therefore, we estimate a UCB of $\mathbb{E}_f[(r-m)^2]/\ln 2$ using a variant of Bernstein's inequality in Proposition 20. This result gives a tentative value of the UCB of s^2 in Lemma 22. Then, applying Lemma 22 to Lemma 17, we obtain Proposition 16.

As is shown in the following lemma, $(r-m)^2$ becomes sub-exponential under the sub-gaussian assumption.

Lemma 19 (Sub-exponential property of square reward) *Let r be an i.i.d. random variable following sub-gaussian $f(r)$ with variance proxy s^2. Then, $\forall m \in \mathbb{R}$, $x = (r-m)^2$ follows a sub-exponential with parameter $b = 2s^2$.*

Proof Let r be an i.i.d. random variable following a sub-gaussian $f(r)$ with mean proxy m and variance proxy s^2. Let $g(x)$ be the distribution of $x = (r-m)^2$. Then, $\forall u \geq 0$,

$$\begin{aligned}\mathbb{P}_g\{x \geq u\} &= \mathbb{P}_f\{(r-m)^2 \geq u\} = \mathbb{P}_f\{|r-m| \geq \sqrt{u}\} \\ &\leq 2\exp\left(-\frac{u}{2s^2}\right) = 2\exp\left(-\frac{u}{b}\right),\end{aligned} \tag{35}$$

where $b = 2s^2$. We used the definition of sub-exponential property for the last inequality. ■

Using the sub-exponential property of $(r-m)^2$, a UCB of $\mathbb{E}_f[(r-m)^2]$ can be estimated from a variant of Bernstein's inequality.

Proposition 20 (Bernstein's inequality) *Let $g(x)$ be a sub-exponential distribution with parameter b. Let $x_1, x_2, \ldots, x_n > 0$ be i.i.d. sub-exponential random variables following $g(x)$. Then,*

$$\mathbb{P}_g\left\{\frac{1}{n}\sum_{i=1}^{n} x_i - \mathbb{E}_g[x] \geq u\right\} \leq \exp\left(2\sqrt{2}nw_+ - nw_+^2 - 2n\right), \tag{36}$$

and

$$\mathbb{P}_g\left\{\mathbb{E}_g[x] - \frac{1}{n}\sum_{i=1}^{n} x_i \geq u\right\} \leq \begin{cases} \exp\left(2\sqrt{2}nw_- - nw_-^2 - 2n\right) & 0 \leq u < 3b/2 \\ \exp\left[-\left(ub^{-1}+1\right)n\right] & 3b/2 \leq u \end{cases}, \tag{37}$$

where $u \geq 0$ and $w_\pm \coloneqq \sqrt{\pm ub^{-1} + 2}$.

The proof is presented in Appendix A. From Lemma 19 and Proposition 20, a UCB of $\mathbb{E}_f[(r-m)^2]$ is given as follows:

Corollary 21 (Confidence bounds of $\mathbb{E}_f[(r-m)^2]$) *Consider n samples $\{r_i\}_{i\in[n]}$ taken from the sub-gaussian distribution $f(r)$ with mean proxy m and variance proxy s^2. Then,*

$$\frac{1}{n}\sum_{i=1}^{n}(r_i-m)^2 - \gamma_+ b \leq \mathbb{E}_f\left[(r-m)^2\right] \leq \frac{1}{n}\sum_{i=1}^{n}(r_i-m)^2 + \gamma_- b, \tag{38}$$

when $\gamma_+ \coloneqq \beta^2 + 2\sqrt{2}\beta$ and

$$\gamma_- \coloneqq \begin{cases} -\beta^2 + 2\sqrt{2}\beta & 0 < \beta < 1/\sqrt{2} \\ 1 + \beta^2 & 1/\sqrt{2} \leq \beta, \end{cases}$$

with a confidence level of $0 < 1-\alpha < 1$, where $\beta \coloneqq \sqrt{-\ln\alpha/n}$ and $b \coloneqq 2s^2$.

Using this corollary, we obtain a tentative UCB of s^2 as follows:

Lemma 22 (Tentative UCB of s^2) *Consider n samples $\{r_i\}_{i\in[n]}$ taken from the sub-gaussian distribution $f(r)$ with mean proxy m and variance proxy s^2. Then,*

$$s^2 \lessapprox \frac{1}{2n(\ln 2 - \gamma)}\sum_{i=1}^{n}(r_i-m)^2 \tag{39}$$

with the confidence level $0 < 1-\alpha < 1 - \exp[-(\sqrt{2}-\sqrt{2-\ln 2})^2 n]$, or almost equivalently $n > -13.613\ln\alpha > 0$. $\gamma \coloneqq -\beta^2 + 2\sqrt{2}\beta$, where $\beta \coloneqq \sqrt{-\ln\alpha/n}$.

Proof Substituting $2s^2 \ln 2 = \mathbb{E}_f[(r-m)^2]$ into Corollary 21,

$$2s^2 \ln 2 \leq \frac{1}{n}\sum_{i=1}^{n}(r_i - m)^2 + 2s^2\gamma_-. \tag{40}$$

Then, when $0 < \gamma_- < \ln 2 < 1/\sqrt{2}$,

$$s^2 \leq \frac{1}{2n(\ln 2 - \gamma_-)}\sum_{i=1}^{n}(r_i - m)^2. \tag{41}$$

From the bounds of γ_-, $0 < \beta < \sqrt{2} - \sqrt{2 - \ln 2}$, which is equivalent to the bounds of α in this lemma. ■

Lemmas 17 and 22 contain unknown $\hat{m}$ and m. We simply select $m = \hat{m} = (\sum_{i=1}^{n} r_i)/n$ because we only estimated the tentative value of $\hat{s}^2$. Proposition 16 is then obtained from these lemmas. Using the sample means for m and $\hat{m}$ is also justified in terms of the order of convergence. The confidence bounds of the sample mean of the reward converge to the expected mean at $O(n^{-1/2})$ (Auer et al., 2002). Then, we expect that m and $\hat{m}$ also converge in the same order. This order is faster than $O(n^{-1/4})$, which is the order of convergence of $\hat{s}$ in Lemma 22. In this case, $I(r; \hat{m}, \hat{s}^2)$ in Lemma 17 converges to $\lim_{n\to\infty} I(r; \hat{m}, \hat{s}^2)$ at the same order of $\hat{s}$. Because of this, it is sufficient to correctly evaluate only $\hat{s}$.

5.2.3 Derivation for Algorithm 3

Setting $\alpha = \nu^{-c^2}$, we can implement the UCB in Theorem 16 as Algorithm 3. We use symbol ν instead of $\tau - 1$ because of the generality in MCTS. In this algorithm, c is a hyperparameter that controls the balance between exploration and exploitation. We recommend $c = 1/\sqrt{13.613}$ to satisfy the condition, $\gamma < \ln 2$, when $n > \ln \nu$. To explore the search space more randomly, a larger c should be used. In the implementation, when the same z value was obtained from other arms, one of the arms was selected randomly. The inequality, $n > \ln \nu$, indicates that our algorithm allocates at least $\ln \nu$ trials for the non-optimal arms. This allocation order is consistent with the optimal order for the conventional bandit problem (Auer et al., 2002).

6. Experiments and Results

We conduct two types of numerical experiments to compare our algorithms with other algorithms. One is the synthetic bandit problems with the Gaussian reward distributions, and the other is SMILES optimization using MCTS (Yang et al., 2017; Kajita et al., 2020; Kikkawa et al., 2020) as the demonstrations for materials discovery. We employed a single set of recommended or reasonable hyperparameters for all the experiments because the tuning of hyperparameters for the actual applications in materials discovery is extremely expensive. We set $T = 10,000$ considering the realistic applications (Kajita et al., 2020; Kikkawa et al., 2020) unless the observed maximum reward clearly does not converge. We present the details of other algorithms in Appendix B.

6.1 Synthetic problems for bandits

The synthetic problems explored in the experiments include the following:

"easy" problem

This problem consists of three arms with the Gaussian parameters $(\mu_1, \sigma_1) = (1, 1)$, $(\mu_2, \sigma_2) = (0, 2)$, and $(\mu_3, \sigma_3) = (-1, 3)$. Arm 3 is optimal for the MKB problem because of its large variance. However, in the conventional bandit approaches, arm 1 is preferred because of its high expectation reward.

"difficult" problem

This problem consists of three arms with $(\mu_1, \sigma_1) = (-0.2, 1.1)$, $(\mu_2, \sigma_2) = (0, 1)$, and $(\mu_3, \sigma_3) = (-0.8, 1.2)$. In this problem, the optimal arm in the MKB problem switches depending on the total number of trials. Arm 1 is optimal $10^2 \ll T \ll 10^9$ because $\mu_1 + 2\sigma_1 = \mu_2 + 2\sigma_2$ and $\mu_1 + 6\sigma_1 = \mu_3 + 6\sigma_3$. The algorithms for determining the arm with the maximum expectation reward will select arm 2. An algorithm with a strong tendency to choose arms with high variances will have a higher preference toward arm 3 than arm 1. It is a challenge for the MKB algorithm to select arm 1 correctly.

"unfavorable" problem

This problem comprises three arms with the same variance; the Gaussian parameters of each arm were set to $(\mu_1, \sigma_1) = (1, 1)$, $(\mu_2, \sigma_2) = (0, 1)$, and $(\mu_3, \sigma_3) = (-1, 1)$, respectively. In this setting, arm 1 is optimal. the conventional UCB will select the optimal arm correctly because this arm has the highest mean reward. The MKB algorithms will lose the conventional UCB because these algorithms incur costs for estimating the variance of each arm.

The transition plots of the observed maximum and the ratio of the optimal arm selection averaged over 100 independent runs are shown in Figure 1. The plots of the observed maximum can directly evaluate the performance of the MKB algorithm; however it is susceptible to data variability. The ratio of the optimal arm selection can help in that case.

In the result of the "easy" problem, the MKB algorithms exhibit higher observed maximum reward than the random search on average. Although the obtained maximum rewards are similar among these MKB algorithms, the ratios of the optimal arm selected clearly show that our algorithms identify the best arm first. As expected, the conventional UCB mainly selected the non-optimal arm. The *spUCB* and *UCBE* also afforded worse results than those of the random search.

In the "difficult" problem, the selection ratios show that the MKB algorithms selected the optimal arm more frequently than the random search, although slight differences were observed in the observed maximum reward. In particular, our algorithms more efficiently select the optimal arm than other MKB algorithms. The performances of the random search and *UCBE* were almost the same, and *spUCB* and the conventional UCB exhibited the worse performances.

In the "unfavorable" case, the conventional UCB worked the best from the viewpoint of the selection ratio. The performance of *spUCB* is similar to that of the conventional UCB. Our algorithms also exhibited good performance, although the ratios were slightly lower than those of the conventional UCB. The results of *ThresholdAscent* and *RobustUCBMax* were better than those of *UCBE*. The random search afforded the worst result.

In the previous three tasks, the *MaxSearch*[*sub-gaussian*] algorithm outperformed the *MaxSearch*[*Gaussian*] algorithm. This performance difference is possibly attributable to different mean estimates in the two algorithms. The Gaussian-based algorithm achieved higher performance when using the sample mean $\bar{\mu}$ than when using the UCB of the mean $\hat{\mu}$.

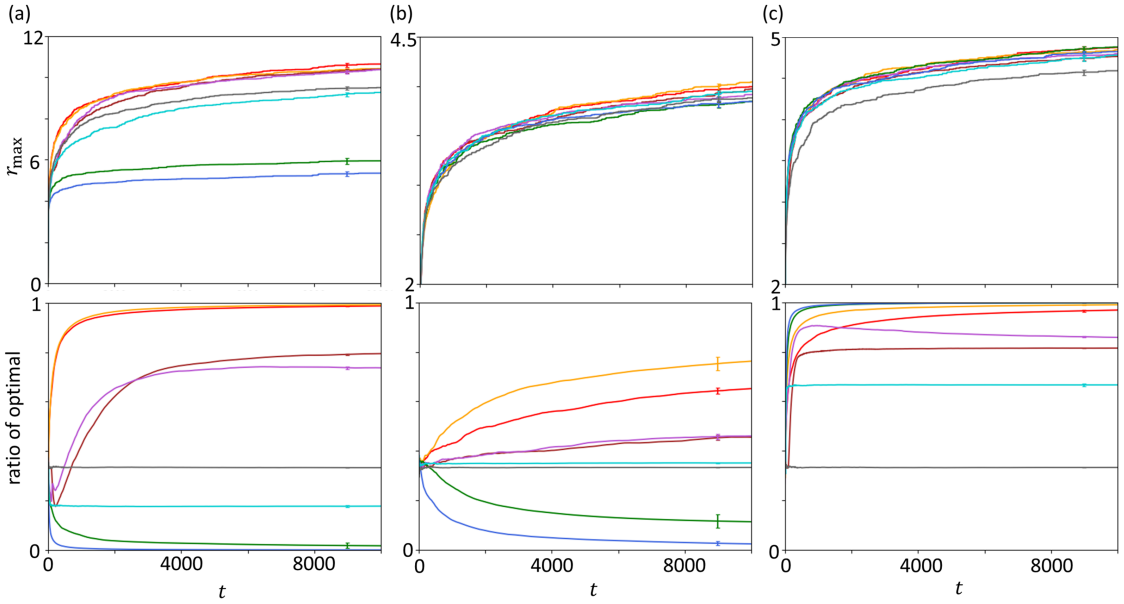

Figure 1: Transition plots of the observed maximum (upper) and ratio of the optimal arm selection (lower) on (a) the "easy" problem, (b) the "difficult" problem, and (c) the "unfavorable" problem. The colors represent the results of different algorithms: red, *MaxSearch*[*Gaussian*]; orange *MaxSearch*[*sub-gaussian*]; reddish brown, *ThresholdAscent*; purple, *RobustUCBMax*; green, *spUCB*; sky blue, *UCBE*; blue, conventional UCB; and gray, random search. The error bars indicate the standard errors of 100 independent runs. If the differences between the two methods are more than two times the standard errors, there will be a significant difference between these methods with a 5 % significance level.

6.2 Molecular discovery using tree search

As a demonstration of the molecular discovery problem, we attempted to optimize the molecular structure M which maximized either of the properties defined by the following empirical equations (Joback and Reid, 1987):

$$T_{\mathrm{b}}(M)[\mathrm{K}] = 198.2 + \sum_{i \in \mathrm{frag}(M)} T_{\mathrm{b},i},$$

$$P_{\mathrm{c}}(M)[\mathrm{bar}] = \left[0.113 + 0.0032 N_{\mathrm{a}}(M) + \sum_{i \in \mathrm{frag}(M)} P_{\mathrm{c},i}\right]^{-2},$$

$$\eta_{300\mathrm{K}}(M)[\mathrm{Pa{\cdot}s}] = M_{\mathrm{w}}(M) \exp\left[\frac{\sum_{i \in \mathrm{frag}(M)} \eta_{\mathrm{a},i} - 597.82}{300} + \sum_{i \in \mathrm{frag}(M)} \eta_{\mathrm{b},i} - 11.202\right],$$

where T_{b}, P_{c}, and $\eta_{300\mathrm{K}}$ are the boiling temperature, critical pressure, and liquid dynamic viscosity at 300 K of molecule M, respectively; $\mathrm{frag}(M)$ was a set of atomic fragments of M, determined by Joback and Raid. The fragments simply determined for each atom type, such as carbon in methyl group, halogens, and ether oxygen in a ring group, etc. The functions $N_{\mathrm{a}}(M)$ and $M_{\mathrm{w}}(M)$ were the number of atoms in M and molecular weight of M, respectively. The empirical parameters, $T_{\mathrm{b},i}$, $P_{\mathrm{c},i}$, $\eta_{\mathrm{a},i}$, and $\eta_{\mathrm{b},i}$, were optimized to reproduce the experimental properties. The properties, T_{b}, P_{c}, and $\eta_{300\mathrm{K}}$, depended on the molecular structure through these parameters. In addition to those three properties, the topological polar surface area $\mathrm{TPSA}(M)$ [Å^2] (Å = 0.1nm) (Ertl et al., 2000) was maximized. Using these empirical formulas, we can verify the performance of the search algorithms in a short time.

During the search process, the candidate molecular structures were generated using the following context-free grammar (Hopcroft et al., 2001) of the SMILES strings (Weininger, 1988). Using the context-free grammar, we could create a simple maze game (Kikkawa et al., 2020) systematically. Here, we applied the following rules:

$$\begin{aligned}
S &\to \mathrm{C}(X)(Y)(Y)(Y), \mathrm{C}(=\mathrm{O})(Y)(Y), \mathrm{C}(Y)\mathrm{C}(Y)(=\mathrm{C}(Y)\mathrm{C}(Y)), \text{or } \mathrm{C}(=\mathrm{O})(\mathrm{O}(Y))(Y),\\
X &\to [\mathrm{H}], \mathrm{F}, \mathrm{Cl}, \mathrm{Br}, \mathrm{C}(X)(Y)(Y), \mathrm{O}(Y), \mathrm{N}(Y)(Y), \mathrm{C}(=\mathrm{O})(Y),\\
&\quad \mathrm{C}(Y)(=\mathrm{C}(Y)(Y)), \text{or } \mathrm{C}(=\mathrm{O})(\mathrm{O}(Y)),\\
Y &\to [\mathrm{H}], \mathrm{F}, \mathrm{Cl}, \mathrm{Br}, \mathrm{C}(X)(Y)(Y), \mathrm{C}(=\mathrm{O})(Y), \mathrm{C}(Y)(=\mathrm{C}(Y)(Y)), \text{or } \mathrm{C}(=\mathrm{O})(\mathrm{O}(Y)),
\end{aligned}$$

where S, X, and Y denote the non-terminal variables, and the upright characters denote the terminals. The start variable is set to S, and a string-generation process is completed when the string no longer has variables. The following additional rule was applied when the number of alphabets was greater than 40:

$$X \text{ or } Y \to [\mathrm{H}].$$

This rule guarantees the termination of the generation process within the moderate molecular size. This limit is approximately 500 g/mol in molecular weight, and most of the known molecules in the database[2] are within the limit. We employed hydrogen as the termination atom, which is commonly used in organic chemistry. The alphabets include the explicit "H", and exclude the parenthesis and equal symbols. The string "Br" and "Cl" are considered as two alphabets. The search space of this molecular generator contains significantly more than 6.248×10^{13} molecular species, which is the number of isomers in $\mathrm{C_{40}H_{82}}$ (Yeh, 1995). We did not consider the synthesizability and the target scope of generated molecules; however, it can be considered by modifying the grammar in practical use.

2. `https://www.rsc.org/Merck-Index/`

The context-free language can be projected to a tree graph (Figure 2). Therefore, the molecular generator can be easily implemented with an MCTS algorithm, as shown in Algorithm 4. The node selection in each layer continues until a complete molecular string is created. Subsequently, the chemical property evaluation is performed, after which the property value is used as the reward. The reward value is recorded in each node passed in the creation, and it is used to calculate the selection indices in the next creation. The complete SMILES strings assigned on the different leaves are treated as the different molecules in this search algorithm even if these molecules have the same molecular symmetries.

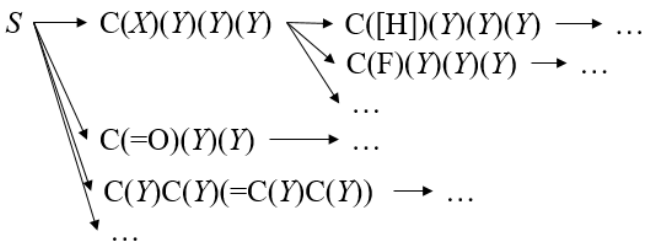

Figure 2: Tree image of SMILES generation.

Algorithm 4 MCTS

Input: number of trials T.

$\tau = 0$
while $\tau < T$ **do**
  $\tau \leftarrow \tau + 1$
  $v \leftarrow root()$ {the root node of the search tree.}
  $L \leftarrow \{v\}$
  while v is not a leaf node **do**
    $k \leftarrow policy(records)$
    {$policy$: Algorithm 1 or the algorithms in Appendix B.
    $records$: statistic data such as K, n_k, R_k, R_k^2, and $r^{\max}$ in Algorithm 1.}
    $L \leftarrow L \cup k$
    $v \leftarrow child(k, v)$ {the k-th child of the node v.}
  end while
  $r \leftarrow reward(L)$ {the reward of the selected path L.}
  $records.add(L, r)$ {record the path L and the reward r.}
end while

The properties, T_b, P_c, and $\eta_{300\mathrm{K}}$ were calculated using the python thermo module (Bell and Contributors, 2016), and TPSA was calculated using the RDKit library (Landrum, 2016). When using $\eta_{300\mathrm{K}}$ as the reward, the rules containing one of F, N, and =C were excluded because their empirical parameters were not available. Additionally, we note that all of the generated SMILES were valid in the network test of RDKit.

Using the transition plots of the observed maximum, we compared *MaxSearch* and other algorithms in Figure 3. The plots were obtained by averaging over each 100 independent search runs.

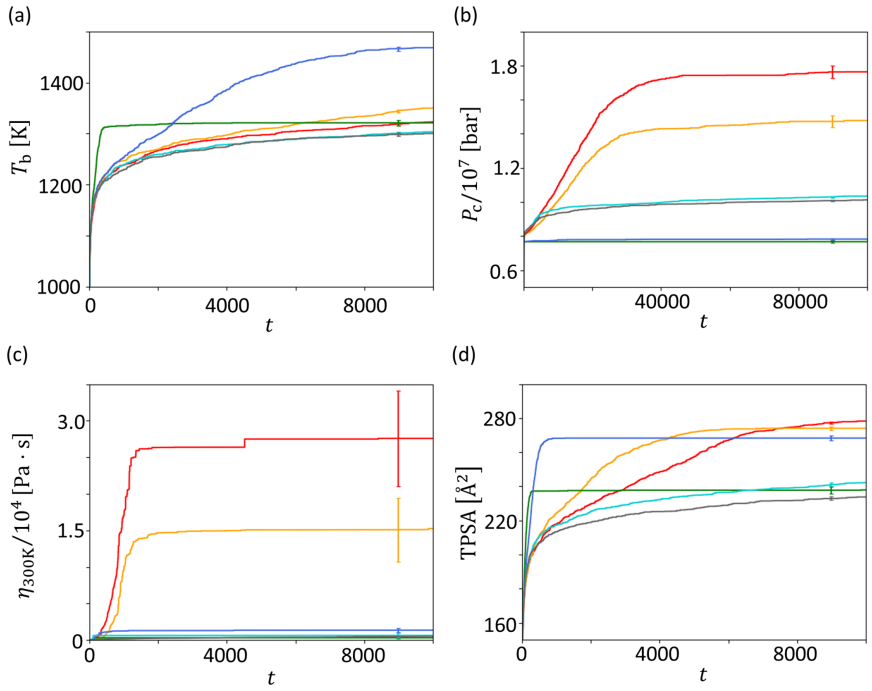

Figure 3: Transition plots of the molecular discovery. (a) T_b, (b) P_c, (c) $\eta_{300\mathrm{K}}$, and (d) TPSA. The other notations are the same as those in Figure 1. The colors represent the results of different algorithms: red, *MaxSearch*[*Gaussian*]; orange, *MaxSearch*[*sub-gussian*]; green, *spUCB*; sky blue, *UCBE*; blue, conventional UCB; and gray, random search. The error bars indicate the standard errors of the 100 independent runs . *ThresholdAscent* and *RobustUCBMax* cannot be implemented in MCTS because of the many hyperparameters involved.

In the search of T_b in Figure 3(a), the conventional UCB afforded the highest rewards at $t = 10,000$. This result is expected because the empirical formula of T_b is a simple sum of the fragment parameters. In such case, the optimal arm is almost equivalent to the arm with the best expectation reward. This condition corresponds to the "unfavorable" case of synthetic problems. In fact, the searches for other properties expressed by simple summation in the Joback method afforded similar results. For $t < 2,500$, *spUCB* demonstrated the best performance. This result is probably due to setting the exploitative hyperparameter $c = 0.1$ recommended in the original article (Schadd et al., 2008). The conventional UCB with $c = 0.1$ gave a similar transition plot. Our algorithms are less effective than the conventional UCB and comparable in performance to the *spUCB*, but outperform *UCBE* and random search.

In the searches of P_c, η_{300K}, and TPSA, our algorithms outperformed the other algorithms, especially in the late stage. There are some different tendencies in these transition plots. These differences are probably due to the differences in the population distributions of rewards. For example, for η_{300K}, there are chemical structures with enormously high rewards in the search space. Our algorithm can find these structures with a high efficiency and success rate. In contrast, for TPSA, the population distribution probably has an upper bound near 290 Å^2. Our algorithms worked well even if such case. These results evidence the wide application range of our proposed algorithms.

The chemical structures with the top three rewards of T_b, P_c, η_{300K}, or TPSA in the 100 independent runs are shown in Figure 4. From the high-scoring molecules, we deduce that:

- Carboxyl groups are favorable for high T_b.
- Alcohol, carboxyl, and halogen groups are favorable for high viscosity.
- Polarized oxygen groups are favorable for high TPSA.

These understandings are consistent with chemical knowledge. More complicated and highly optimized structures can be found in our algorithms than other algorithms.

7. Discussion

The numerical experiments in the previous section show that the UCB of EI is possible as the selection index for the MKB problem. In this section, we discuss why that is so. Our discussion would contain non-rigorous arguments. However, we believe that this discussion will help with future work. We also discuss the incompleteness of our algorithm for multimodal reward distributions. This limitation is due to the sub-gaussian assumption, but this would not affect the application of MCTS.

7.1 Subtleties of Extreme Regret

For our discussion, we should mention the subtleties of extreme regret, first pointed out by Nishihara et al. (2016). In this section, we review these subtleties.

The extreme regret was introduced by Carpentier and Valko (2014), defined as follows:

Definition 23 (Carpentier's regret) *In the MKB problem, Carpentier's regret when $k(t)$, $t \in [T]$ are selected is defined as follows:*

$$R_C^{k_*(t),k(t)}(T) \coloneqq \mathbb{E}\left[\max_{t\in[T]} r_{k_*(t)}(t)\right] - \mathbb{E}\left[\max_{t\in[T]} r_{k(t)}(t)\right], \tag{42}$$

where $k_*(t), t \in [T]$ denotes an oracle policy. The asymptotically optimal policy should satisfy

$$R_C^{k_*(t),k(t)}(T) = o\left(\mathbb{E}\left[\max_{t\in[T]} r_{k_*(t)}(t)\right]\right). \tag{43}$$

A subtlety of Carpentier's regret is that the regret asymptotically approaches 0 for most policies in some settings. For example, we consider all reward distributions of the arms have bounded support. Then, any policy that selects each arm infinitely often achieves an

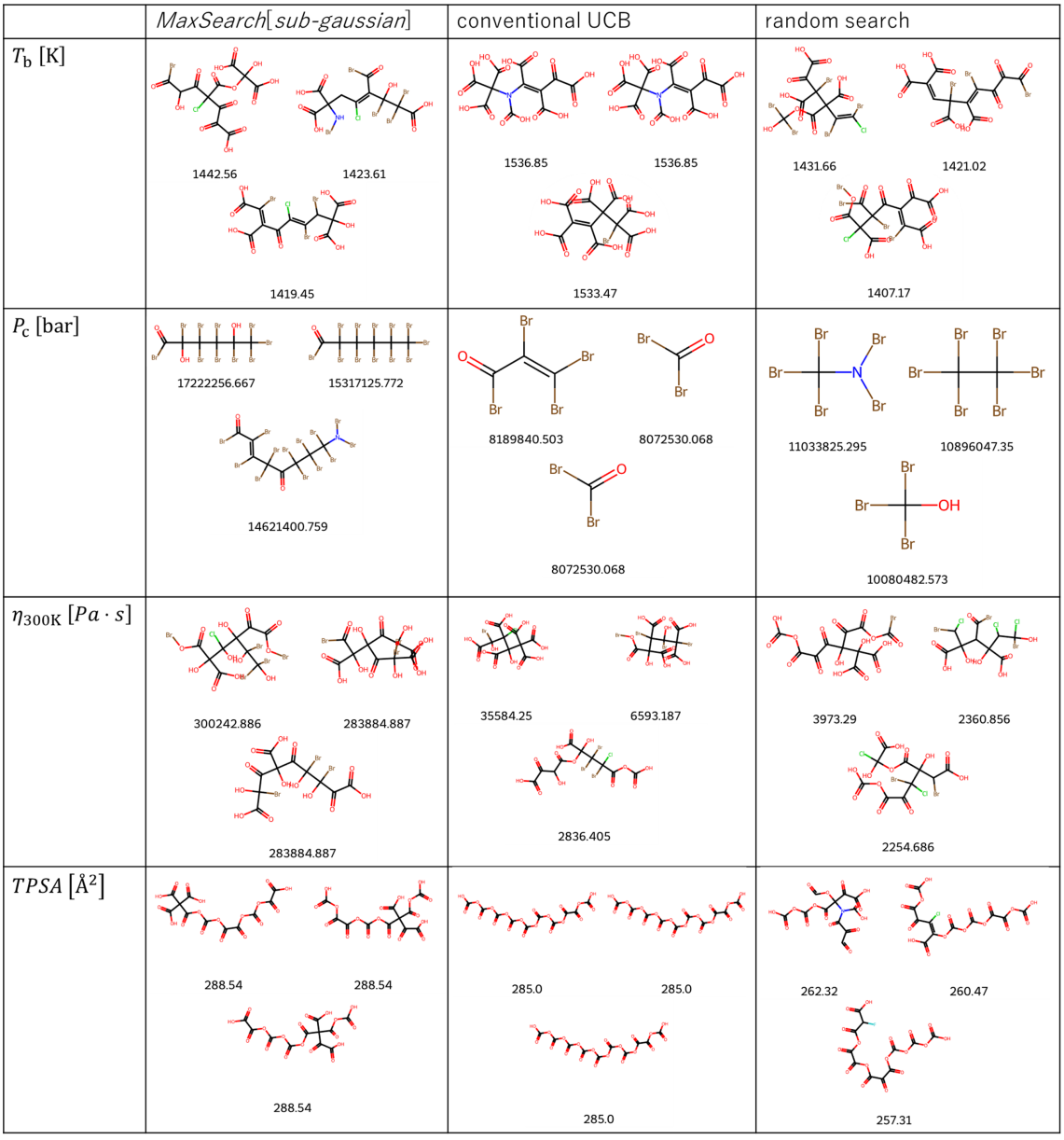

Figure 4: Chemical structures with the top three rewards in 100 independent runs.

asymptotically zero regret, meaning that even the random search is asymptotically optimal in the setting. To avoid this, Nishihara et al. (2016) defined an alternative regret:

Definition 24 (Nishihara's regret) *In the MKB problem, Nishihara's regret when $k(t)$, $t \in [T']$ are selected is defined as follows:*

$$R_N^{k_*(t),k(t)}(T) \coloneqq \frac{1}{T} \min_{T' \geq 1} \left\{ T' : \mathbb{E}\left[\max_{t \in [T']} r_{k(t)}(t) \right] \geq \mathbb{E}\left[\max_{t \in [T]} r_{k_*(t)}(t) \right] \right\}, \tag{44}$$

where $k_*(t), t \in [T]$ denotes an oracle policy. The asymptotically optimal policy should satisfy

$$\limsup_{T \to \infty} R_N^{k_*(t),k(t)}(T) \leq 1. \tag{45}$$

This regret works even when the reward distributions have bounded supports. However, Nishihara et al. (2016) showed that there is a set of reward distributions such that

$$\limsup_{T \to \infty} R_{\mathrm{N}}^{k_*^{\mathrm{SA}},k(t)}(T) \geq K$$

for any policy, where k_*^{SA} is the selection of single-armed oracle defined in Definition 25. Namely, no policy is asymptotically optimal under Nishihara's regret. Because of this, we only treat reward distributions with unbounded supports in the following discussion.

Another subtlety exists in the definition of the oracle policy. The previous works are essentially based on the single-armed oracle (Nishihara et al., 2016) as follows:

Definition 25 (Single-armed oracle) *In the MKB problem, the single-armed oracle is the policy that plays the single arm*

$$k_*^{SA}(T) := \mathrm{argmax}_{k \in [K]} \mathbb{E}\left[\max_{t \in [T]} r_k(t) \right] \tag{46}$$

over a time horizon T.

However, this oracle gives different k_*^{SA} depending on T. This fact can be confirmed by the following example.

Example 1 *Consider the MKB problem with $K = 3$. Let the reward distributions of each arm be $f_1(r) = \mathcal{N}(r; 0, 0.01)$, $f_2(r) = \mathcal{N}(r; -1, 0.25)$, and $f_3(r) = \mathcal{N}(r; -15, 4)$. Then, the single-armed oracle gives $k_*^{SA} = 1$ if $T \leq 11$, $k_*^{SA} = 2$ if $1.4 \times 10^{11} \leq T \leq 5.4 \times 10^{13}$, and $k_*^{SA} = 3$ if $T \geq 3.9 \times 10^{202}$.*

Proof The expected maximum reward sampled from the k-th arm over a time horizon T is bounded by

$$\mu_k + \frac{1}{\sqrt{\pi \ln 2}} \sigma_k \sqrt{\ln T} \leq \mathbb{E}\left[\max_{t \in [T]} r_k(t) \right] \leq \mu_k + \sqrt{2} \sigma_k \sqrt{\ln T}, \tag{47}$$

where μ_k and σ_k^2 are the mean and variance of the Gaussian reward distribution, $f_k(r)$, respectively (Kamath, 2015). Then, the example is established. ■

The T-dependency of the single-armed oracle means that the best arm cannot be determined without information on T (Nishihara et al., 2016). This raises a question about the regret analysis using the infinity limit of T. In Example 1, arm 3 should be selected most often to achieve the asymptotically zero regret. However, arm 1 or 2 is a more suitable choice when $T < 5.4 \times 10^{13}$. Because many applications cannot perform such a large number of trials, the regret analysis result is impractical. The T-dependency of the oracle is also inconvenient in MCTS applications. In an MCTS algorithm, T is not given except for the root node. Then, one cannot determine the best arm except for the root node even if the reward distribution is known.

7.2 T-independent oracles and asymptotics of UCB approach

Nishihara et al. (2016) also proposed an oracle independent of T.

Definition 26 (Nishihara's greedy oracle) *In the MKB problem, Nishihara's greedy oracle is the policy that plays the arm with the maximum EI. Namely, this oracle plays*

$$k_*^N(\tau) := \underset{k\in[K]}{\operatorname{argmax}}\, \mathbb{E}\left[\max\left\{r_k(\tau), r_*^{max}(\tau-1)\right\} - r_*^{max}(\tau-1) | \{r_{k_*^N(t)}(t)\}_{t\in\tau-1}\right] \tag{48}$$

at time τ, where $r_*^{max}(\tau) \coloneqq \max_{t\in[\tau]} r_{k_*^N}(t)$.

This oracle uses EI to avoid the dependency on T. Therefore, in terms of Nishihara's greedy oracle, it is natural that we employ EI to derive a T-independent MKB algorithm. Although Nishihara et al. (2016) did not analyze this oracle much, we note that Nishihara's greedy oracle gives different $k_*^{\mathrm{N}}(\tau)$ depending on the oracle value $r_*^{\max}(\tau)$ instead of T, as shown in the following example:

Example 2 *Consider the MKB problem with $K = 3$. Let the reward distributions of each arm be $f_1(r) = \mathcal{N}(r; 0, 1)$, $f_2(r) = \mathcal{N}(r; -2, 2)$, and $f_3(r) = \mathcal{N}(r; -6, 3)$. Then, Nishihara's greedy oracle gives $k_*^N(\tau) = 1$ if $r_*^{max}(\tau-1) \leq 1.3$, $k_*^N(\tau) = 2$ if $7.0 \leq r_*^{max}(\tau-1) \leq 11.9$, and $k_*^N(\tau) = 3$ if $r_*^{max}(\tau-1) \geq 18.9$,*

Proof Because of Lemma 14, we should consider the integral of the survival function. The survival function of $f_k(r) = \mathcal{N}(r; \mu_k, \sigma_k^2)$ is expressed as follows:

$$S(r) \coloneqq \int_r^\infty f_k(u)du = \frac{1}{2}\,\mathrm{erfc}\left[\frac{r-\mu_k}{2\sigma_k}\right] \tag{49}$$

The bounds of $\mathrm{erfc}(x), x > 0$ are given by

$$c\exp(-\beta x^2) < \mathrm{erfc}(x) < \exp(-x^2), \tag{50}$$

where

$$c = \sqrt{\frac{2e}{\pi}}\frac{\sqrt{\beta-1}}{\beta}, \tag{51}$$

and $\beta > 1$ (Chiani et al., 2003; Chang et al., 2011). Then,

$$\frac{c}{2}\int_y^\infty \exp(-\beta x^2)dx < \frac{1}{2}\int_y^\infty \mathrm{erfc}(x)dx < \frac{1}{2}\int_y^\infty \exp(-x^2)dx, \tag{52}$$

where $y > 0$. Using the definition and bounds of $\mathrm{erfc}(x)$ again, we obtain

$$\frac{\sqrt{\pi}c^2}{4\sqrt{\beta}} \exp(-\beta^2 y^2) < \frac{1}{2}\int_y^\infty \mathrm{erfc}(x)dx < \frac{\sqrt{\pi}}{4}\exp(-y^2). \tag{53}$$

These bounds give the example. ■

This dependency generates a subtlety in an adaptive case. Consider one obtains $r^{\max}(\tau) < 11.9$ under a selection $\{k(t)\}_{t\in[\tau]}$ in Example 2. A problem arises when the oracle value $r_*^{\max}(\tau) > 18.9$ at that time. In this case, the next selection of Nishihara's greedy oracle differs from the selection with the maximum EI, meaning that a policy simply approaching Nishihara's greedy oracle is not always effective in the MKB problem.

The subtlety due to the dependence on the oracle value $r_*^{\max}(\tau)$ is solved using the observed value $r^{\max}(\tau)$ alternatively. We define this oracle as follows:

Definition 27 (Kikkawa's greedy oracle) *In the MKB problem, let $k(t), t \in [\tau - 1]$ be the previous selections. Then, Kikkawa's greedy oracle plays*

$$k_*^K(\tau) := \underset{k\in[K]}{\mathrm{argmax}}\, \mathbb{E}\left[\max\left\{r_k(\tau), r^{max}(\tau-1)\right\} - r^{max}(\tau-1) | \{r_{k(t)}(t)\}_{t\in\tau-1}\right] \tag{54}$$

at time τ, where $r^{max}(\tau) \coloneqq \max_{t\in[\tau]} r_{k(t)}(t)$.

This oracle is equivalent to Nishihara's greedy oracle when all selections follow this oracle. In addition, this oracle gives the arm that has the maximum EI even when the non-oracle selections exist in $k(t), t \in [\tau - 1]$. The following proposition states that Kikkawa's greedy oracle asymptotically approaches Nishihara's greedy oracle in terms of Carpentier's regret.

Proposition 28 (Asymptotics of Kikkawa's greedy oracle) *Let $k(t), t \in [T]$ contain $o(T)$ non-oracle selections and other selections follow Kikkawa's greedy oracle. Then,*

$$R_C^{k_*^N(t),k(t)}(T) = o\left(\mathbb{E}\left[\max_{t\in[T]} r_{k_*^N(t)}(t)\right]\right), \tag{55}$$

when

$$EI\left[k, T; G^{max}\right] = O\left(\frac{1}{T}\mathbb{E}\left[\max_{t\in[T]} r_{k_*^N(t)}(t)\right]\right). \tag{56}$$

Proof Let $n_k(T), k \in [K]$ and $n_k^{\mathrm{N}}(T), k \in [K]$ be the numbers of the k-th arm selected in $k(t), t \in [T]$ and $k_*^{\mathrm{N}}(t), t \in [T]$, respectively. Then, $\delta n \coloneqq \left|n_k^{\mathrm{N}}(T) - n_k(T)\right| = o(T)$ is

expected.[3] Therefore,

$$\begin{aligned}\left|R_{\mathrm{C}}^{k_*^{\mathrm{N}}(t),k(t)}(T)\right| &= \left|\mathbb{E}\left[\max_{k\in[K]}\max_{n\in[n_k^{\mathrm{N}}(T)]} r_k(n)\right] - \mathbb{E}\left[\max_{k\in[K]}\max_{n\in[n_k(T)]} r_k(n)\right]\right| \\ &= \left|\mathbb{E}\left[\max_{k\in[K]}\left[\max\left\{\max_{n\in[n^{\min}]} r_k(n), \max_{n\in[\delta n]} r_k(n+n^{\min})\right\} - \max_{n\in[n^{\min}]} r_k(n)\right]\right]\right| \\ &\leq \sum_{k\in[K]}\left|\mathbb{E}\left[\max\left\{\max_{n\in[n^{\min}]} r_k(n), \max_{n\in[o(T)]} r_k(n+n^{\min})\right\} - \max_{n\in[n^{\min}]} r_k(n)\right]\right| \\ &\leq \sum_{k\in[K]}\left|o(T)EI\left[k, n^{\min}; G^{\max}\right]\right| = o\left(\mathbb{E}\left[\max_{t\in[T]} r_{k_*^{\mathrm{N}}(t)}(t)\right]\right),\end{aligned} \tag{57}$$

where $n^{\min} \coloneqq \min\{n_k^{\mathrm{N}}(T), n_k(T)\}$. ■

The proposition states that $o(T)$ mistakes are allowed in the asymptotically optimal policy under the condition related to the maximum value. This condition can be satisfied by Gaussian distributions at least.

Our concept in Section 4 can be obtained by simply substituting the EI in Kikkawa's greedy oracle into its UCB. Therefore, our conceptual algorithm is expected to approach Kikkawa's greedy oracle for large T. The number of non-oracle selections in Algorithm 1 can be estimated as follows:

Proposition 29 (Number of non-oracle selections) *Consider the MKB problem. Let* $\overline{EI}\left[k, \mathcal{R}(t-1); G^{max}\right], t \in [T]$ *be an estimator of* $EI\left[k, t; G^{max}\right]$. *Suppose a confidence interval of* $EI\left[k, t; G^{max}\right]$ *is known as*

$$\left|EI\left[k, t; G^{max}\right] - \overline{EI}\left[k, \mathcal{R}(t-1); G^{max}\right]\right| \leq C(k, \alpha(t), n_k(t)) \tag{58}$$

with confidence level $1-\alpha(t)$, *where* $n_k(t), k \in [K]$ *be the numbers of the* k*-th arm selected under Algorithm 1 with this confidence interval. Then, the number of non-oracle selections becomes* $o(T)$ *when* $\alpha(t) = o(1)$ *and*

$$C(k, \alpha(t), n_k(t)) = o\left(\left\{\min_{k'\in\kappa(t)} \Delta_{k'}(t)\right\}\left\{\frac{t}{n_k(t)}\right\}^d\right) \tag{59}$$

where $d > 0$, $\kappa(t) = [K]/k_*^K(t)$ *and*

$$\Delta_k(t) = EI\left[k_*^K(t), t; G^{max}\right] - EI\left[k, t; G^{max}\right]. \tag{60}$$

Proof Consider the following events:

$$A_{k_*^{\mathrm{K}}(t),t} : \overline{EI}\left[k_*^{\mathrm{K}}(t), \mathcal{R}(t-1); G^{\max}\right] + C(k_*^{\mathrm{K}}(t), \alpha(t), n_{k_*^{\mathrm{K}}}(t)) \geq EI\left[k_*^{\mathrm{K}}(t), t; G^{\max}\right], \tag{61}$$

$$A_{\kappa(t),t} : \overline{EI}\left[\kappa(t), \mathcal{R}(t-1); G^{\max}\right] \leq EI\left[\kappa(t), t; G^{\max}\right] + C\left(\kappa(t), \alpha(t), n_{\kappa(t)}(t)\right). \tag{62}$$

3. Pathological conditions may exist. However, we do not consider them.

Then, the number of complementary cases can easily be counted as follows:

$$\sum_{t\in[T]}\mathbb{E}\left[\mathbb{1}\left[\bigcup_{k\in[K]}A_{k,t}^{c}\right]\right]\leq K\sum_{t\in[T]}\alpha(t)=o(T). \tag{63}$$

Conversely, in the case of all $A_{k,t}$ established, the non-oracle arm is selected when

$$\begin{aligned}\overline{EI}\left[\kappa(t),\mathcal{R}(t-1);G^{\max}\right]+C(\kappa(t),\alpha(t),n_{\kappa(t)}(t))\\ \geq\overline{EI}\left[k_*^{\mathrm{K}}(t),\mathcal{R}(t-1);G^{\max}\right]+C(k_*^{\mathrm{K}}(t),\alpha(t),n_{k_*^{\mathrm{K}}(t)}(t))\end{aligned} \tag{64}$$

for any $\kappa(t)$. Then,

$$\begin{aligned}EI\left[k_*^{\mathrm{K}}(t),t;G^{\max}\right]&\leq\overline{EI}\left[k_*^{\mathrm{K}}(t),\mathcal{R}(t-1);G^{\max}\right]+C(k_*^{\mathrm{K}}(t),\alpha(t),n_{k_*^{\mathrm{K}}(t)}(t))\\ &\leq\overline{EI}\left[\kappa(t),\mathcal{R}(t-1);G^{\max}\right]+C(\kappa(t),\alpha(t),n_{\kappa(t)}(t))\\ &\leq EI\left[\kappa(t),t;G^{\max}\right]+2C(\kappa(t),\alpha(t),n_{\kappa(t)}(t)).\end{aligned} \tag{65}$$

Equations (61), (64), and (62) are used in the first, second, and third inequalities, respectively. Then, solving for $n_{\kappa(t)}(t)$ using Equation (59), we obtain $n_{\kappa(t)}(t)=o(t)$. Then, we obtain the proposition using Equation (63). ■

This proposition means that Algorithm 1 asymptotically approaches Kikkawa's greedy oracle in terms of the number of non-oracle selections if true UCB was used. That is, Algorithm 1 is conceptually also an asymptotically optimal policy in terms of Nishihara's greedy oracle and Carpentier's regret through Proposition 28.

Under Proposition 29, the *MaxSearch*[*Gaussian*] algorithm corresponds to the following case:

$$\overline{EI}\left[k,\mathcal{R}(t-1);G^{\max}\right]=\sqrt{\frac{\bar{\sigma}_k^2}{2}}\ \mathrm{ierfc}\left(\frac{r-\bar{\mu_k}}{\sqrt{2\bar{\sigma_k}^2}}\right) \tag{66}$$

and

$$C(k,\alpha(t),n_k(t))=O\left(\left\{\frac{\ln t}{n_k(t)}\right\}^{\frac{1}{4}}\right). \tag{67}$$

The derivation is shown in Appendix C. The function $C(k,\alpha(t),n_k(t))$ satisfies the condition in Proposition 29. The algorithm is then asymptotically optimal if the reward of each arm follows the Gaussian distribution.

Similarly, *MaxSearch*[*sub-gaussian*] corresponds to the case with

$$\overline{EI}\left[k,\mathcal{R}(t-1);G^{\max}\right]=I(r^{\max};\hat{m}_k,\lim_{\alpha\to0}\hat{s}_k^2) \tag{68}$$

and

$$C(k,\alpha(t),n_k(t))=O\left(\left\{\frac{\ln t}{n_k(t)}\right\}^{\frac{1}{4}}\right) \tag{69}$$

as the upper bound. Note that no lower bounds are stated in Theorem 16. Because the lower bound of $EI\left[k,t;G^{\max}\right]$ is required in Proposition 29, incorrect cases can occur. An example given below.

Example 3 *Consider the two-armed bandit problem with the reward distributions,* $f_1(r) = Bernoulli(0.5), r \in \{0, 1\}$ *and* $f_2(r) = \mathcal{N}(r; 0, 0.1)$*. Then,* $I(r^{max}; \hat{m}_1, \hat{s}_1^2) > I(r^{max}; \hat{m}_2, \hat{s}_2^2)$ *in most situations. It means that MaxSearch[sub-gaussian] selects arm* 1 *mainly. However, the oracle policy selects arm* 2 *after obtaining* $r = 1$ *once from arm* 1 *because* $EI[1, t; G^{max}] = 0$ *and* $EI[2, t; G^{max}] > 0$ *in this case. This result is not due to the bounded support of the Bernoulli distribution. The similar discussion is established even if the Bernoulli distribution is convolved with* $\mathcal{N}(r; 0, 0.01)$*.*

The algorithm failure in this example is due to the poor estimating ability of the sub-gaussian assumption for the lower bound of $EI[1, t; G^{\max}]$ in Proposition 29. In our algorithm, the tail distribution should approach the estimator at $n_k(t) \gg (\ln t)^4$. The multimodal distribution, such as the Bernoulli distribution, can break this condition easily and has a bad effect for the optimal arm selection. Fortunately, these effects have not impacts empirically in our materials discovery demonstrations using MCTS. We infer that this is a feature of MCTS, and its theoretical analysis is an interesting issue for future studies.

Lastly, we note that the proof of Proposition 29 is an analog of the proof for the conventional bandit problem (Auer et al., 2002; Jamieson, 2018) except for the optimal arm depending on the time t. This treatment can be allowed because the selections by Kikkawa's greedy oracle correspond to the arms with the maximum EI for any t. This feature of Kikkawa's greedy oracle is valuable. We are sure that several other proofs for the conventional bandit problem establish formally in the MKB problem using Kikkawa's greedy oracle.

8. Conclusion

Here, we proposed MKB algorithms and applied them to synthetic problems and molecular-design demonstrations using MCTS for materials discovery. The proposed algorithms use a single hyperparameter and are easily implemented for MCTS. This feature gives the proposed algorithms an advantage over other MKB algorithms, and enables their application to materials discovery. In fact, to the best of our knowledge, this is the first case where the MKB algorithms are actually employed for materials discovery. The performances of the proposed algorithms were examined on the synthetic problems and molecular-structure optimizations. The experimental results demonstrated that the proposed algorithms found the maximum reward more efficiently than other algorithms when the optimal arm could not be determined only based on the expectation reward. In real molecular designs, most of the molecular properties would have a high complexity; thus, we believe that the MKB algorithms are useful for these tasks.

In the theoretical aspect, we mainly contribute in two aspects. One is the proof of the effectiveness of the use of a UCB of EI. The proof result has wide flexibility, and this can be used to propose other algorithms with other assumptions for distributions, which will be addressed in future work. Especially, we indicate that our algorithm is weak for multimodal distributions because it is based on the Gaussian or sub-gaussian reward assumption. Nonparametric approaches are probably required for the MKB algorithm to show their true potential. Recent studies on the nonparametric MKB algorithms (Bhatt et al., 2022; Baudry et al., 2022) probably have a potential to overcome these difficulties. In another direction, theoretical analyses based on full parametric assumptions would help

us to thoroughly understand the MKB problem. The other contribution is the proposal of Kikkawa's greedy oracle. Using the proposed oracle, we can avoid many of the subtleties of the MKB problem. We think that several other statements in the conventional bandit problem can be imported to the MKB problem with the help of this oracle. It may be possible to apply our theoretical approach to other fields based on the bandit problem.

Heuristics to reduce the required trials are also important for actual use. For this purpose, the MKB algorithm could combined with other bandit algorithms. Combining with a supervised learning also holds significant promise.

Acknowledgments

We thank Dr. Ryosuke Jinnouchi in TCRDL for reviewing our early draft. We also thank Nanako Ishihara for her internship work. We also thank the anonymous reviewers. Their incisive remarks have contributed to the theoretical quality of our manuscript.

Appendix A. Proof of Theorem 20

Let X_i be independent random variables drawn from the same sub-exponential $g(X)$ with parameter $b > 0$. Then,

$$\begin{aligned}
&\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}_{g(X)}\left[X\right]\right| \geq u\right\} \\
&= \mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}_{g(X)} \geq u \text{ or } \mathbb{E}_{g(X)} - \mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} X_i\right\} \geq u\right\},
\end{aligned} \tag{70}$$

where $u > 0$. Therefore, in the former case,

$$\begin{aligned}
&\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}_{g(X)}\left[X\right] \geq u\right\} \\
&= \mathbb{P}\left\{\exp\left(\frac{\lambda}{n}\sum_{i=1}^{n} X_i\right) \geq \exp\left[\lambda(u + \mathbb{E}_{g(X)}\left[X\right])\right]\right\} \\
&\leq \exp\left[-\lambda(u + \mathbb{E}_{g(X)}\left[X\right])\right]\mathbb{E}\left[\exp\left(\frac{\lambda}{n}\sum_{i=1}^{n} X_i\right)\right], \quad \langle \text{Markov}'\text{s inequality}\rangle
\end{aligned} \tag{71}$$

where $\lambda > 0$ is an arbitrary parameter. Since the random variables X_i are independent of each other, their moment-generating function can be separated. Thus, we obtain

$$\begin{aligned}
&\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}_{g(X)}\left[X\right] \geq u\right\} \\
&\leq \exp\left[-\lambda(u + \mathbb{E}_{g(X)}\left[X\right])\right]\left\{\mathbb{E}_{g(X)}\left[\exp\left(\frac{\lambda X}{n}\right)\right]\right\}^n \\
&= \exp\left[-\lambda(u + \mathbb{E}_{g(X)}\left[X\right])\right]\left(1 + \frac{\lambda\mathbb{E}_{g(X)}\left[X\right]}{n} + \sum_{p=2}^{\infty}\frac{\lambda^p\mathbb{E}_{g(X)}\left[X^p\right]}{n^p p!}\right)^n \\
&\leq \exp\left[-\lambda u + \sum_{p=2}^{\infty}\frac{\lambda^p\mathbb{E}_{g(X)}\left[X^p\right]}{n^{p-1}p!}\right] \quad \langle \text{Since } 1 + x \leq \exp x\rangle \\
&= \exp\left[-\lambda u + \sum_{p=2}^{\infty}\frac{\lambda^p}{n^{p-1}p!}\int_0^{\infty}\mathbb{P}_{g(X)}\{X^p \geq u\}\,du\right] \quad \langle \text{Integral identity}\rangle \\
&= \exp\left[-\lambda u + \sum_{p=2}^{\infty}\frac{\lambda^p}{n^{p-1}p!}\int_0^{\infty}\mathbb{P}_{g(X)}\{X \geq bv\}pb^pv^{p-1}\,dv\right] \quad \langle \text{Replace } u \text{ with } b^pv^p\rangle \\
&\leq \exp\left[-\lambda u + 2\sum_{p=2}^{\infty}\frac{\lambda^p b^p}{n^{p-1}(p-1)!}\int_0^{\infty}e^{-v}v^{p-1}\,dv\right]. \quad \langle \text{Sub-exponential}\rangle
\end{aligned} \tag{72}$$

The above integral corresponds to the Gamma function. Therefore,

$$\begin{aligned}\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}X_i-\mathbb{E}_{g(X)}[X]\geq u\right\}&\leq\exp\left[-\lambda u+2\sum_{p=2}^{\infty}\frac{b^p\lambda^p\Gamma(p)}{n^{p-1}(p-1)!}\right]\\&=\exp\left[-\lambda u+2\sum_{p=2}^{\infty}\frac{b^p\lambda^p}{n^{p-1}}\right]\\&=\exp\left[-\lambda u+\frac{2b^2\lambda^2}{n-b\lambda}\right],\end{aligned}\tag{73}$$

where $|b\lambda/n|<1$. Replacing $n-b\lambda$ with ξ_+, we obtain

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}X_i-\mathbb{E}_{g(X)}[X]\geq u\right\}\leq\exp\left[\left(\frac{u}{b}+2\right)\xi_+ +\frac{2n^2}{\xi_+}-\frac{nu}{b}-4n\right],\tag{74}$$

where $0<\xi_+<2n$. Hence, the optimized ξ_+ is

$$\xi_+=\sqrt{\frac{2n^2b}{2b+u}}.\tag{75}$$

Then,

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}X_i-\mathbb{E}_{g(X)}[X]\geq u\right\}\leq\exp\left(2\sqrt{2}nw_+-nw_+^2-2n\right),\tag{76}$$

where $w_+\coloneqq\sqrt{ub^{-1}+2}\geq\sqrt{2}$. Moreover, using the same approach, we obtain

$$\mathbb{P}\left\{\mathbb{E}_{g(X)}[X]-\frac{1}{n}\sum_{i=1}^{n}X_i\geq u\right\}\leq\exp\left[\left(-\frac{u}{b}+2\right)\xi_- +\frac{2n^2}{\xi_-}+\frac{nu}{b}-4n\right],\tag{77}$$

where $\xi_-\coloneqq n+b\lambda$ and $0<\xi_-<2n$. Hence, the optimized ξ_- is

$$\xi_-=\begin{cases}\sqrt{2n^2b/(2b-u)} & 0\leq u<3b/2\\ 2n-\epsilon & 3b/2\leq u,\end{cases}\tag{78}$$

where ϵ is an infinitesimal. Then, we obtain

$$\mathbb{P}\left\{\mathbb{E}_{g(X)}[X]-\frac{1}{n}\sum_{i=1}^{n}X_i\geq u\right\}\leq\begin{cases}\exp\left(2\sqrt{2}nw_--nw_-^2-2n\right) & 0\leq u<3b/2\\ \exp\left[\left(-ub^{-1}+1\right)n+O(\epsilon)\right] & 3b/2\leq u,\end{cases}\tag{79}$$

where $w_-\coloneqq\sqrt{-ub^{-1}+2}>1/\sqrt{2}$. Equations 76 and 79 show that $\mathbb{E}[X]$ is bounded at a confidence level of $1-\alpha>0$ as follows:

$$\frac{1}{n}\sum_{i=1}^{n}X_i-u_+^*\leq\mathbb{E}[X]\leq\frac{1}{n}\sum_{i=1}^{n}X_i+u_-^*,\tag{80}$$

where

$$\alpha=\exp\left[2\sqrt{2}nw_+^*-n(w_+^*)^2-2n\right],\tag{81}$$

when $u_+^* \geq 0$, and

$$\alpha = \begin{cases} \exp\left[2\sqrt{2}nw_-^* - n(w_-^*)^2 - 2n\right] & 0 \leq u_-^* < 3b/2 \\ \exp\left[\left(-u_-^* b^{-1} + 1\right) n\right] & 3b/2 \leq u_-^*, \end{cases} \tag{82}$$

where $w_\pm^* \coloneqq \sqrt{\pm u_\pm^* b^{-1} + 2}$ (double sign in the same order). From Eq. 81, we obtain

$$w_+^* = \sqrt{2} + \beta, \tag{83}$$

and

$$u_+^* = (\beta^2 + 2\sqrt{2}\beta)b, \tag{84}$$

where $\beta \coloneqq \sqrt{-\ln \alpha / n}$. In addition, from Eq. 82,

$$u_-^* = \begin{cases} (-\beta^2 + 2\sqrt{2}\beta)b & 0 \leq \beta < 1/\sqrt{2} \\ (\beta^2 + 1)b & 1/\sqrt{2} \leq \beta. \end{cases} \tag{85}$$

The theorem follows from Eqs. 76 and 79.

Appendix B. Compared Algorithms

We compared our algorithm with Algorithms 5-10. We employed the following hyperparameters and applied some modifications for the implementation. In *ThresholdAscent*, the hyperparameters were set to $s = 100$ and $\delta = 2 \ln \nu$. We used the reward ranking instead of the iteration used in the original code (Streeter and Smith, 2006b). In *RobustUCBMax*, we set $s = 100$, $u = r^{s\text{-th}}$, $v = (r^{\max} - u)^{1+\epsilon}/\sqrt{\nu}$, and $\epsilon = 0.4$, according to the original paper (Achab et al., 2017). Although the original paper employed the robust UCB with the truncated mean estimator, we used a simple version of the robust UCB (Bubeck et al., 2013). In *spUCB*, $c = 0.1$ and $D = 32$ are used as the hyperparameters. These values are recommended in the original paper (Schadd et al., 2008). In *UCBE* (Audibert et al., 2010) and the conventional UCB (Auer et al., 2002), we used $c = 1$ as the hyperparameter. In some algorithms, we estimated the variance parameter, σ, as the sample variance of the first $P = 10$ random searches.

Algorithm 5 ThresholdAscent

Input: number of arms K, time horizon T, the s-th maximum of observed reward $r^{s\text{-th}}$ number of times the k-th arm is selected n_k, the i-th reward from the k-th arm $r_{k,i}$, and hyperparameter δ.

Output: selected arm index $\hat{k}$.

for each $k \in [K]$ **do**
  if $n_k = 0$ or $\nu < 2$ **then**
    $z_k \leftarrow \infty$
  else
    $S_k = \sum_{i \in [n_k]} \mathbb{1}[r_{k,i} > r^{s\text{-th}}]$
    $\alpha \leftarrow \ln(2TK/\delta)$
    $z_k \leftarrow S_k/n_k + (\alpha + \sqrt{\alpha(2S_k + \alpha)})/n_k$
  end if
end for
$\hat{k} \leftarrow \underset{k \in [K]}{\operatorname{argmax}}\, z_k$
return $\hat{k}$

Algorithm 6 RobustUCBMax

Input: number of arms K, number of times the k-th arm is selected n_k, the i-th reward from the k-th arm $r_{k,i}$, and hyperparameters u, v, and ϵ.

Output: selected arm index $\hat{k}$.

$\nu = \sum_{k \in [K]} n_k$
for each $k \in [K]$ **do**
  if $n_k = 0$ or $\nu < 2$ **then**
    $z_k \leftarrow \infty$
  else
    $S_k = \sum_{i \in [n_k]} r_{k,i} \mathbb{1}[r_{k,i} > u]$
    $z_k \leftarrow S_k/n_k \ + 4v^{1/(1+\epsilon)} (2 \ln \nu / n_k)^{\epsilon/(1+\epsilon)}$
  end if
end for
$\hat{k} \leftarrow \underset{k \in [K]}{\operatorname{argmax}}\, z_k$
return $\hat{k}$

Algorithm 7 spUCB

Input: number of arms K, current time τ, number of times the k-th arm is selected n_k, sum of the rewards obtained from the k-th arm R_k, sum of square rewards obtained from the k-th arm Q_k, and sample variance obtained from the first P trials σ.

Output: selected arm index $\hat{k}$.

if $\tau \leq P$ **then**
  $\hat{k} \leftarrow RandomSearch(K)$
else
  $\nu = \sum_{k \in [K]} n_k$
  for each $k \in [K]$ **do**
    if $n_k = 0$ or $\nu < 2$ **then**
      $z_k \leftarrow \infty$
    else
      $m_k \leftarrow R_k / n_k$
      $z_k \leftarrow m_k + c\sigma\sqrt{\ln \nu / n_k} \ + \sqrt{\frac{Q_k - n_k m_k^2 + D}{n_k}}$
    end if
  end for
  $\hat{k} \leftarrow \underset{k \in [K]}{\operatorname{argmax}}\, z_k$
end if
return $\hat{k}$

Algorithm 8 UCBE

Input: number of arms K, current time τ, number of times the k-th arm is selected n_k, sum of the rewards obtained from the k-th arm R_k, and sample variance obtained from the first P trials σ.

Output: selected arm index $\hat{k}$.

if $\tau \leq P$ **then**
  $\hat{k} \leftarrow RandomSearch(K)$
else
  $\nu = \sum_{k \in [K]} n_k$
  for each $k \in [K]$ **do**
    if $n_k = 0$ or $\nu < 2$ **then**
      $z_k \leftarrow \infty$
    else
      $z_k \leftarrow R_k / n_k + c\sigma\sqrt{\nu / n_k}$
    end if
  end for
  $\hat{k} \leftarrow \underset{k \in [K]}{\operatorname{argmax}}\, z_k$
end if
return $\hat{k}$

Algorithm 9 UCB

Input: number of arms K, current time τ, number of times the k-th arm is selected n_k, sum of the rewards obtained from the k-th arm R_k, and sample variance obtained from the first P trials σ.

Output: selected arm index $\hat{k}$.

if $\tau \leq P$ **then**
  $\hat{k} \leftarrow RandomSearch(K)$
else
  $\nu = \sum_{k \in [K]} n_k$
  for each $k \in [K]$ **do**
    if $n_k = 0$ or $\nu < 2$ **then**
      $z_k \leftarrow \infty$
    else
      $z_k \leftarrow R_k / n_k + c\sigma\sqrt{\ln \nu / n_k}$
    end if
  end for
  $\hat{k} \leftarrow \underset{k \in [K]}{\operatorname{argmax}}\, z_k$
end if
return $\hat{k}$

Algorithm 10 RandomSearch

Input: number of arms K.

Output: selected arm index $\hat{k}$.

$\hat{k} \leftarrow random(K)$ {randomly select any of $1, ..., K$.}
return $\hat{k}$

Appendix C. Order of $C(k, \alpha(t), n)$

In the Gaussian reward settings, we have

$$C(k, \alpha(t), n) = \sqrt{\frac{\hat{\sigma}_k^2}{2}}\ \text{ierfc}\left(\frac{r - \hat{\mu}_k}{\sqrt{2\hat{\sigma}_k^2}}\right) - \sqrt{\frac{\bar{\sigma}_k^2}{2}}\ \text{ierfc}\left(\frac{r - \bar{\mu}_k}{\sqrt{2\bar{\sigma}_k^2}}\right). \tag{86}$$

The order of the above equation can be derived using some asymptotic equations (Zelen and Sevelo, 1968). For large n and $\left|\mathcal{N}_{\alpha/2}\right| \ll \sqrt{2(n-1)}$, we can write

$$\hat{\mu}_k = \bar{\mu}_k + t_{n-1,1-\alpha/2}\frac{\bar{\sigma}_k}{\sqrt{n}} \approx \bar{\mu}_k + \mathcal{N}_{1-\alpha/2}[1 + O(n^{-1})]\frac{\bar{\sigma}_k}{\sqrt{n}} \tag{87}$$

$$\hat{\sigma}_k^2 = \frac{(n-1)\bar{\sigma}_k^2}{\chi^2_{n-1,\alpha/2}} \approx \frac{2(n-1)\bar{\sigma}_k^2}{(\mathcal{N}_{\alpha/2} + \sqrt{2n-3})^2} \approx \bar{\sigma}_k^2 - \frac{\mathcal{N}_{\alpha/2}}{\sqrt{2(n-1)}} \tag{88}$$

$$\mathcal{N}_{\frac{1}{2}\pm\frac{1-\alpha}{2}} = \Theta(\sqrt{-\ln\alpha}), \tag{89}$$

where $\mathcal{N}_p$ is the p-quantile of the standard normal distribution. Therefore, if $\alpha = \nu^{-c^2}$,

$$\left|\mathcal{N}_{\alpha/2}\right| \ll \sqrt{2(n-1)} \Leftrightarrow \ln\nu \ll n, \tag{90}$$

$$\mathcal{N}_{\frac{1}{2}\pm\frac{1-\alpha}{2}} = \Theta(\sqrt{\ln\nu}) \tag{91}$$

$$\hat{\mu}_k - \bar{\mu}_k = \Theta\left(\sqrt{\frac{\ln\nu}{n}}\right) \tag{92}$$

$$\hat{\sigma}_k^2 - \bar{\sigma}_k^2 = \Theta\left(\sqrt{\frac{\ln\nu}{n}}\right). \tag{93}$$

Because $\ln\nu \ll n$, we can apply the Taylor expansion to $C(k, \alpha(t), n)$ as follows:

$$C(k, \alpha(t), n) = O(\hat{\mu}_k - \bar{\mu}_k) + O(\hat{\sigma}_k - \bar{\sigma}_k) = O\left(\left\{\frac{\ln\nu}{n}\right\}^{\frac{1}{4}}\right). \tag{94}$$

The order of $C(k, \alpha(t), n)$ under sub-gaussian reward settings can also be derived through Taylor expansion and the order of $\hat{s}_k^2$ in Proposition 16.

$$C(k, \alpha(t), n) = I(r; \hat{m}_k, \hat{s}_k^2) - I(r; \bar{m}_k, \bar{s}_k^2) = O\left(\left\{\frac{\ln\nu}{n}\right\}^{\frac{1}{4}}\right). \tag{95}$$

References

Mastane Achab, Stephan Clémençon, Aurélien Garivier, Anne Sabourin, and Claire Vernade. Max k-armed bandit: On the extremehunter algorithm and beyond. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 389–404. Springer, 2017.

Ankit Agrawal and Alok Choudhary. Deep materials informatics: Applications of deep learning in materials science. MRS Communications, 9(3):779–792, 2019.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In COLT, pages 41–53. Citeseer, 2010.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2):235–256, 2002.

Dorian Baudry, Yoan Russac, and Emilie Kaufmann. Efficient algorithms for extreme bandits. arXiv preprint arXiv:2203.10883, 2022.

Caleb Bell and Contributors. Thermo: Chemical properties component of chemical engineering design library (chedl). 2016. URL `https://github.com/CalebBell/thermo`.

Sujay Bhatt, Ping Li, and Gennady Samorodnitsky. Extreme bandits using robust statistics. IEEE Transactions on Information Theory, 2022.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games, 4(1):1–43, 2012.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Gábor Lugosi. Bandits with heavy tail. IEEE Transactions on Information Theory, 59(11):7711–7717, 2013.

Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. Nature, 559(7715):547–555, 2018.

Alexandra Carpentier and Michal Valko. Extreme bandits. In Neural Information Processing Systems, 2014.

Seok-Ho Chang, Pamela C Cosman, and Laurence B Milstein. Chernoff-type bounds for the gaussian error function. IEEE Transactions on Communications, 59(11):2939–2944, 2011.

Marco Chiani, Davide Dardari, and Marvin K Simon. New exponential bounds and approximations for the computation of error probability in fading channels. IEEE Transactions on Wireless Communications, 2(4):840–845, 2003.

Vincent A Cicirello and Stephen F Smith. The max k-armed bandit: A new model of exploration applied to search heuristic selection. In The Proceedings of the Twentieth National Conference on Artificial Intelligence, volume 3, pages 1355–1361, 2005.

Yahel David and Nahum Shimkin. PAC lower bounds and efficient algorithms for the max k-armed bandit problem. In International Conference on Machine Learning, pages 878–887. PMLR, 2016.

Zachary Del Rosario, Matthias Rupp, Yoolhee Kim, Erin Antono, and Julia Ling. Assessing the frontier: Active learning, model accuracy, and multi-objective candidate discovery and optimization. The Journal of Chemical Physics, 153(2):024112, 2020.

Peter Ertl, Bernhard Rohde, and Paul Selzer. Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. Journal of Medicinal Chemistry, 43(20):3714–3717, Oct 2000. ISSN 0022-2623. doi: 10.1021/jm000942e. URL https://doi.org/10.1021/jm000942e.

John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Introduction to automata theory, languages, and computation, 2nd edition. SIGACT News, 32(1):60–65, March 2001. ISSN 0163-5700. doi: 10.1145/568438.568455. URL https://doi.org/10.1145/568438.568455.

Kevin Jamieson. Lecture 3: Stochastic multi-armed bandits, regret minimization, 2018. URL https://courses.cs.washington.edu/courses/cse599i/18wi/resources/lecture3/lecture3.pdf.

Dipendra Jha, Logan Ward, Arindam Paul, Wei-keng Liao, Alok Choudhary, Chris Wolverton, and Ankit Agrawal. Elemnet: Deep learning the chemistry of materials from only elemental composition. Scientific reports, 8(1):1–13, 2018.

Dipendra Jha, Kamal Choudhary, Francesca Tavazza, Wei-keng Liao, Alok Choudhary, Carelyn Campbell, and Ankit Agrawal. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. Nature communications, 10(1):1–12, 2019.

K. G. Joback and R. Reid. Estimation of pure-component properties from group-contributions. Chemical Engineering Communications, 57:233–243, 1987.

Shenghong Ju, TM Dieb, K Tsuda, and J Shiomi. Optimizing interface/surface roughness for thermal transport. In Machine Learning for Molecules and Materials NIPS 2018 Workshop, 2018.

Seiji Kajita, Tomoyuki Kinjo, and Tomoki Nishi. Autonomous molecular design by Monte-Carlo tree search and rapid evaluations using molecular dynamics simulations. Communications Physics, 3(1):1–11, 2020.

Gautam Kamath. Bounds on the expectation of the maximum of samples from a gaussian. URL http://www. gautamkamath. com/writings/gaussian max. pdf, 2015.

Nobuaki Kikkawa, Seiji Kajita, and Kensuke Takechi. Self-learning molecular design for high lithium-ion conductive ionic liquids using maze game. Journal of Chemical Information and Modeling, 60(10):4904–4911, 2020.

Shin Kiyohara and Teruyasu Mizoguchi. Searching the stable segregation configuration at the grain boundary by a monte carlo tree search. The Journal of chemical physics, 148 (24):241741, 2018.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In European conference on machine learning, pages 282–293. Springer, 2006.

A Gilad Kusne, Heshan Yu, Changming Wu, Huairuo Zhang, Jason Hattrick-Simpers, Brian DeCost, Suchismita Sarker, Corey Oses, Cormac Toher, Stefano Curtarolo, et al. On-the-fly closed-loop materials discovery via Bayesian active learning. Nature communications, 11(1):1–11, 2020.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. Advances in applied mathematics, 6(1):4–22, 1985.

Greg Landrum. RDKit: Open-source cheminformatics software. 2016. URL `https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_04`.

Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. Materials discovery and design using machine learning. Journal of Materiomics, 3(3):159–177, 2017. ISSN 2352-8478. doi: https://doi.org/10.1016/j.jmat.2017.08.002.

Thaer M. Dieb, Shenghong Ju, Kazuki Yoshizoe, Zhufeng Hou, Junichiro Shiomi, and Koji Tsuda. MDTS: automatic complex materials design using Monte Carlo tree search. Science and technology of advanced materials, 18(1):498–503, 2017.

Thaer M. Dieb, Zhufeng Hou, and Koji Tsuda. Structure prediction of boron-doped graphene by machine learning. The Journal of chemical physics, 148(24):241716, 2018.

Bryce Meredig, Erin Antono, Carena Church, Maxwell Hutchinson, Julia Ling, Sean Paradiso, Ben Blaiszik, Ian Foster, Brenna Gibbons, Jason Hattrick-Simpers, Apurva Mehta, and Logan Ward. Can machine learning identify the next high-temperature superconductor? examining extrapolation performance for materials discovery. Mol. Syst. Des. Eng., 3:819–825, 2018. doi: 10.1039/C8ME00012C.

Robert Nishihara, David Lopez-Paz, and Léon Bottou. No regret bound for extreme bandits. In Artificial Intelligence and Statistics, pages 259–267. PMLR, 2016.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. Journal of cheminformatics, 9(1):1–14, 2017.

Frank WJ Olver, Daniel W Lozier, Ronald F Boisvert, and Charles W Clark. NIST handbook of mathematical functions hardback and CD-ROM. Cambridge university press, 2010.

Tarak K Patra, Troy D Loeffler, and Subramanian KRS Sankaranarayanan. Accelerating copolymer inverse design using monte carlo tree search. Nanoscale, 12(46):23653–23662, 2020.

Ghanshyam Pilania, Chenchen Wang, Xun Jiang, Sanguthevar Rajasekaran, and Ramamurthy Ramprasad. Accelerating materials property predictions using machine learning. Scientific reports, 3(1):1–6, 2013.

H. Pishro-Nik. Introduction to Probability, Statistics, and Random Processes. Kappa Research, LLC, 2014. ISBN 9780990637202. URL `https://books.google.co.jp/books?id=3yq_oQEACAAJ`.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. Science advances, 4(7):eaap7885, 2018.

Paul Raccuglia, Katherine C Elbert, Philip DF Adler, Casey Falk, Malia B Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A Friedler, Joshua Schrier, and Alexander J Norquist. Machine-learning-assisted materials discovery using failed experiments. Nature, 533 (7601):73–76, 2016.

Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. Machine learning in materials informatics: recent applications and prospects. npj Computational Materials, 3(1):1–13, 2017.

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. Science, 361(6400):360–365, 2018.

Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L Guimaraes, and Alan Aspuru-Guzik. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). ChemRxiv, 2017, 2017.

Maarten PD Schadd, Mark HM Winands, H Jaap Van Den Herik, Guillaume MJ-B Chaslot, and Jos WHM Uiterwijk. Single-player monte-carlo tree search. In International Conference on Computers and Games, pages 1–12. Springer, 2008.

Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic AI. Nature, 555(7698):604–610, 2018.

Matthew J Streeter and Stephen F Smith. An asymptotically optimal algorithm for the max k-armed bandit problem. In AAAI, pages 135–142, 2006a.

Matthew J Streeter and Stephen F Smith. A simple distribution-free approach to the max k-armed bandit problem. In International Conference on Principles and Practice of Constraint Programming, pages 560–574. Springer, 2006b.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

Tsuyoshi Ueno, Trevor David Rhone, Zhufeng Hou, Teruyasu Mizoguchi, and Koji Tsuda. COMBO: an efficient Bayesian optimization library for materials science. Materials discovery, 4:18–21, 2016.

Roman Vershynin. High-dimensional probability: An introduction with applications in data science, volume 47. Cambridge university press, 2018.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences, 28(1):31–36, Feb 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005. URL https://pubs.acs.org/doi/abs/10.1021/ci00057a005.

Hironao Yamada, Chang Liu, Stephen Wu, Yukinori Koyama, Shenghong Ju, Junichiro Shiomi, Junko Morikawa, and Ryo Yoshida. Predicting materials properties with little data using shotgun transfer learning. ACS central science, 5(10):1717–1730, 2019.

Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. ChemTS: an efficient python library for de novo molecular generation. Science and technology of advanced materials, 18(1):972–976, 2017.

Chin-yah Yeh. Isomer enumeration of alkanes, labeled alkanes, and monosubstituted alkanes. Journal of chemical information and computer sciences, 35(5):912–913, 1995.

Marvin Zelen and Norman C. Sevelo. Probability functions. In Milton Abramowitz and Irene A Stegun, editors, Handbook of mathematical functions with formulas, graphs, and mathematical tables, chapter 26, pages 925–996. US Government printing office, 1968.