A Bandit Approach with Evolutionary Operators for Model Selection

Margaux Brégère, Julie Keisler

A BANDIT APPROACH WITH EVOLUTIONARY OPERATORS FOR MODEL SELECTION: APPLICATION TO NEURAL ARCHITECTURE OPTIMIZATION FOR IMAGE CLASSIFICATION

Margaux Brégère
EDF Lab Paris-Saclay
University of Paris, CNRS, LPSM
margaux.bregere@edf.fr

Julie Keisler
EDF Lab Paris-Saclay
University of Lille & INRIA
julie.keisler@edf.fr

ABSTRACT

This paper formulates model selection as an infinite-armed bandit problem. The models are arms, and picking an arm corresponds to a partial training of the model (resource allocation). The reward is the accuracy of the selected model after its partial training. In this best arm identification problem, regret is the gap between the expected accuracy of the optimal model and that of the model finally chosen. We first consider a straightforward generalization of UCB-E (see Audibert et al., 2010) to the stochastic infinite-armed bandit problem and show that, under basic assumptions, the expected regret order is $T^{-\alpha}$ for some $\alpha \in (0, 1/5)$ and T the number of resources to allocate. From this vanilla algorithm, we introduce the algorithm Mutant-UCB that incorporates operators from evolutionary algorithms. Tests carried out on three open source image classification data sets attest to the relevance of this novel combining approach, which outperforms the state-of-the-art for a fixed budget.

***K*eywords** Infinite-armed bandits · Model selection · Neural architecture optimisation · Hyperparameter optimisation · Evolutionnary algorithm · Image classification · AutoML · Online Learning

1 Introduction

Accuracy of machine learning models significantly depends on some parameters which cannot be modified during training. As the number of parameter combinations to be tested exponentially increases with the number of these parameters, it becomes costly and time-consuming to optimize them. Automating the selection of promising models, usually referred to as AutoML (Automated Machine Learning), is a fast-growing area of research (see Hutter et al., 2019 for a quite recent book). We approach the model selection problem in a general manner without any restrictions on the nature of the hyper-parameters, such as the types of machine learning models, neural network architectures, or hyper-parameters of random forests. Our aim is to find the best model without making any assumptions about the task, model type, or reward to maximize. We assume that we have access to an infinite number of possible models. To accomplish this, we set a predetermined budget of resources T, used to train the models. These resources are allocated to the models in the form of "sub-trains", such as iterations, data samples or features. The final model is chosen by finding a good trade-off between exploration (training a large number of models) and exploitation (allocating a large budget to promising models). This process may fall under the umbrella of bandits (see Lattimore and Szepesvári, 2020 for an in-depth review).

In this paper, we treat model selection as an instance of best-arm identification in infinite-armed bandits and consider Upper Confidence Bound (UCB)-based algorithms (see Auer et al., 2002). From these we developed a new model selection algorithm, called Mutant-UCB by adding a mutation operator from the Evolutionary Algorithm to UCB-E (see Audibert et al., 2010). This operator creates a new model from the neighborhood of the model selected by the bandit algorithm. Unlike most model selection algorithms, Mutant-UCB makes no assumptions about the solutions encoding, also called search space, or the reward function to be maximized, making it suitable for a wide range of

configurations. The use of a UCB-type algorithm and adaptive resource allocation allows exploration of the search space, while the mutation operator effectively directs the search towards promising solutions. Results on a neural networks optimization problem demonstrate the relevance of this approach.

We begin this paper by presenting the setup of our bandit model selection approach in Section 2 and we position ourselves in relation to the state of the art. A generalization of UCB-E (see Audibert et al., 2010) to the stochastic infinite-armed bandit problem, called ∞-UCB-E, and an upper regret bound for this algorithm are proposed in Section 3. Based on ∞-UCB-E, we developped a new algorithm called Mutant-UCB, which is presented Section 4. Section 5 is dedicated to experiments on the optimization of deep neural networks. We validate the performance of Mutant-UCB on three open-source image classification data sets. Finally, Section 6 discusses the advantages of Mutant-UCB compared to the state of the art and opens new research perspectives.

2 Model selection problem setup: a bandit approach

2.1 Literature Discussion

Naive strategies for model selection are grid or random search. More sophisticated strategies address model selection as a sequential learning problem. Two approaches stand out: **configuration selection** methods sequentially select new models ("close" to promising models) to train, while **configuration evaluation** [Li et al., 2018] methods allocate more resources (training time) to promising models. The first approach suggests the existence of a distance between models (namely an underlying space), so that two models close to each other will have similar performance, while the second comes with no assumption about the potential (smooth)-links between model performances.

Evolutionary methods. Among the configuration selection methods, evolutionary algorithms have been popular for many years (see, e.g., Young et al., 2015 and Jian et al., 2023). Starting from an initial set of configurations, they evolve them towards performing models using unitary operators like the mutation (little change in the configuration), or even complex operators involving more than two configurations like the crossover Strumberger et al., 2019. These algorithms are highly versatile and can be applied to a wide range of setups. The literature presents different methods that vary in terms of search spaces, i.e. the way configurations are encoded. The operators used to generate the new population typically depend on this encoding. Usually, the configurations are represented as character strings or lists and can be modified using bit-string mutations and combined with k-point crossovers (see Eiben and Smith, 2015 for more details). But recent works, mostly for neural networks architecture optimization, tried to design other representations and operators. For instance a tree-based mutation operator to optimize recurrent neural networks is proposed by Rawal and Miikkulainen [2018]. Awad et al. [2021] use differential evolutionary operators to optimize neural network hyperparameters and architectures. One disadvantage of the evolutionary algorithms is the large number of parameters involved, such as the population size, the selection function, or the elitism rate. Choosing the appropriate values for these parameters can be complex.

Bayesian optimization. Bayesian optimization has recently emerged as a more efficient approach than evolutionary methods in AutoML (see, among others, Malkomes et al., 2016, Zoph and Le, 2016 and Kandasamy et al., 2018). It is a sequential optimization technique commonly used to minimize black-box functions. Those algorithms are based on two main components, a surrogate model that approximates the unknown black-box function, and an acquisition function that selects the next element in the search space to be evaluated. One major limitation of these acquisition functions is their reliance on strong assumptions about the black-box function and the search space Garrido-Merchán and Hernández-Lobato [2020]. Therefore, we did not employ a Bayesian optimization algorithm in our experiments as we aimed to avoid making any assumptions about the smoothness, distance or continuity of the search space or the reward function.

Bandits approaches. Firstly, still in the field of Bayesian optimization, the extensions GP-UCB and KernelUCB (see, Srinivas et al., 2009, Valko et al., 2013, respectively) of the classical UCB bandit algorithm and more recent algorithms largely inspired by them (see, e.g., Dai et al., 2023) have been massively used for optimization and eventually model selection. BayesGap algorithm introduced by Hoffman et al. [2014] connects Bayesian optimization approaches and best arm identification, assuming correlations among the arms. More recently, Huang et al. [2021] sees the neural architecture search as a combinatorial multi-armed bandit problem which allows the decomposition of a large search space into smaller blocks where tree-search methods can be applied more effectively and efficiently. Configuration evaluation approaches have also been investigated in an infinite or multi-armed bandit framework. At each iteration of the algorithm, a new arm/model can be drawn from an infinite search space containing the models and added to the set of models already (more or less) trained. Karnin et al. [2013] proposes the Sequential (or Successive) Halving algorithm, which splits the given budget evenly across an optimal number of elimination rounds, and within a round,

pulls arms in a uniform manner. It comes with solid theoretical guarantees that have recently been improved by Zhao et al. [2023]. Li et al. [2018] proposes the algorithm Hyperband, a robust extension of Sequential Halving, and applies it to deep neural networks hyperparameters optimization. Moreover, Shang et al. [2019] introduces D-TTTS, an algorithm inspired by Thompson sampling. Hybrid methods combine adaptive configuration selection and evaluation: Terayama et al. [2021] proposes a rule to stop training a model prematurely based on the predicted performance from Gaussian Process; in addition, Kandasamy et al. [2016] extends GP-UCB to enable sequential model training (and thus resource allocation).

Best-arm identification in infinite-armed bandits. The stochastic infinite-armed bandit framework have been introduced and studied for the cumulative reward maximization problem by Berry et al. [1997] and Wang et al. [2008]. Carpentier and Valko [2015] and Aziz et al. [2018] study best armed identification problem in this framework. They prove that their strategies (SiRI and extensions; α,ϵ-KL-LUCB, respectively) are minimax optimal either up to a log factor.

2.2 Contributions

In this section we first consider the ∞-UCB-E algorithm. This is a straightforward generalization of UCB-E [Audibert et al., 2010] to stochastic infinite-armed bandits. We show that, under basic assumptions, the expected regret order for this algorithm is $T^{-\alpha}$ with $\alpha \in (0, 1/5)$. This result helps us to set the appropriate number of models to sample from the search space before running UCB-E. Hyperband and Successive halving algorithms generally outperform UCB-based approaches. Besides, algorithms cited in the previous paragraph outperform the regret bound of the ∞-UCB-E algorithm (see, e.g, Karnin et al., 2013). However, this algorithm is the basis of our second and main contribution and the regret bound is a first milestone to attest the relevance of our approach.

From this vanilla algorithm, we propose the algorithm Mutant-UCB that incorporates techniques from evolutionary algorithms. It combines both configuration evaluation and configuration selection approaches: it is sequential in computation and picks a (generally promising) model thanks to a UCB-based criteria. Then it either continues its training (resource allocation) or creates and starts training a new model derived from the selected one thanks to the "mutation" operation of an evolutionary algorithm. This last possibility is based on the intuition that the expected "mutant model" accuracy will be close to that of the original model. While bandit approaches have been used to design the "selection" operator for evolutionary algorithms(see Li et al., 2013), to our knowledge this is the first time that operators from evolutionary algorithms are incorporated into a bandit algorithm.

Finally, we compare Mutant-UCB to a random search, the evolutionary algorithm proposed by Keisler et al. [2023] and Hyperband (see Li et al., 2018) on three open source data sets collected for image classification: CIFAR-10 [Krizhevsky et al., 2009], MRBI [Larochelle et al., 2007] and SVHN [Netzer et al., 2011]. For a fair comparison, Mutant-UCB and the evolutionary algorithm under consideration share the same mutation operation.

2.3 Set-up

When a new arm k is pulled from the search space, the expectation of the accuracy of the associated model μ_k is assumed to be an independent sample from a fixed distribution. With T a fixed budget, at each round $t = 1, \dots, T$, an arm I_t is picked and a sub-train (allocation of a resource) is performed on the associated model. This model is then evaluated on a validation data set $\mathcal{D}_{\text{valid}}$ using an accuracy function acc we aim to maximize. This accuracy corresponds to the reward a_t. Conditionally to the chosen arm, the reward is assumed independent from the past and so we have $\mathbb{E}[\, a_t \,|\, I_t = k] = \mu_k$. In practice, it is highly likely that successive sub-trains will greatly improve accuracy, thereby invalidating the assumption of independence between rewards of the same model. After T rounds, we select the final arm $\widehat{I}_T$. In this best-arm identification problem in infinite-armed bandit framework; if we let $\mu^\star \in \mathrm{argmax}_{k \in \mathbb{N}^\star} \mu_k$ be the accuracy expectation of the best possible model, we aim to minimize the regret

$$r_T = \mu^\star - \mu_{\widehat{I}_T} \,. \tag{1}$$

In what follows, the untrained model associated with arm k is denoted f_k, and after N_k sub-trains we denote it $f_k^{N_k}$.

3 An extension of UCB-E for infinite-armed bandits

3.1 Algorithm

Algorithm 1 introduces ∞-UCB-E, a variant of the UCB-E algorithm proposed by Audibert et al. [2010] in which the number K of arms (or, in our use case, untrained models) is not given by the bandit problem but a parameter of the

Algorithm 1 ∞-UCB-E

Inputs:
T budget
E exploration parameter
K number of untrained models
Initialization
Sample K untrained models $f_1, \ldots, f_K$
For $k = 1, 2, \ldots, K$
Perform a first sub-train on f_k which becomes f_k^1
Get the reward $a_k = \mathrm{acc}\big(f_k^1, \mathcal{D}_{\mathrm{valid}}\big)$
Define $N_k = 1$, $\hat{\mu}_k = a_k$
For $t = K+1, K+2, \ldots, T$
Choose $I_t \in \mathrm{argmax}_{k \in \{1,\ldots K\}} \left\{ \hat{\mu}_k + \sqrt{\frac{E}{N_k}} \right\}$
Perform a sub-train: model $f_{I_t}^{N_{I_t}}$ becomes $f_{I_t}^{N_{I_t}+1}$
Get the reward $a_t = \mathrm{acc}\big(f_{I_t}^{N_{I_t}+1}, \mathcal{D}_{\mathrm{valid}}\big)$
Update $\hat{\mu}_{I_t} = \frac{1}{N_{I_t}+1}\big(r_t + N_{I_t}\hat{\mu}_{I_t}\big)$ and $N_{I_t} = N_{I_t} + 1$
Output:
Model $f_{\widehat{I}_T}^{N_{\widehat{I}_T}}$ where $\widehat{I}_T \in \mathrm{argmax}_{k \in \{1,\ldots K\}} \hat{\mu}_k$

algorithm. This highly exploratory policy is based on the principle of optimism in the face of uncertainty, in the spirit of the UCB algorithm introduced by Auer et al. [2002]. Once K has been fixed and the untrained models $f_1, \ldots, f_K$ have been sampled from the search space, the algorithm starts by K rounds of deterministic exploration: it performs a first sub-train per model and observes the accuracies $a_k = \mathrm{acc}\big(f_k^1, \mathcal{D}_{\mathrm{valid}}\big)$. At each round $t = K+1, \ldots, T$, and for each k, it computes the empirical mean accuracy $\hat{\mu}_k$ from the previous rewards associated with arm k:

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{s=1}^{t-1} a_t \mathbf{1}_{I_s = k} \quad \text{with} \quad N_k = \sum_{s=1}^{t-1} \mathbf{1}_{I_s = k} \,. \tag{2}$$

Then, it chooses the arm optimistically:

$$I_t \in \underset{k \in \{1,\ldots K\}}{\mathrm{argmax}} \left\{ \hat{\mu}_k + \sqrt{\frac{E}{N_k}} \right\}, \tag{3}$$

performs a sub-train on the associated model and receives the reward $a_t = \mathrm{acc}\big(f_{I_t}^{N_{I_t}+1}, \mathcal{D}_{\mathrm{valid}}\big)$. The core issue is the tuning of the exploration parameter E. Audibert et al. [2010] show that the optimal value depends on the difficulty of the underlying bandit problem, which has no reason to be known in advance. We will see below that, under some basic assumptions on the distribution of the expectations $\mu_1, \ldots, \mu_k, \ldots$, setting $E = \frac{25}{36}\frac{T-K}{K^5}$ with K of order T^α, where $\alpha \in (0, 1/5)$, is a good choice to minimize the regret expectation.

3.2 Upper bound on the simple regret expectation

Unlike UCB-E, it is almost surely impossible in our case to select the best arm and achieve the maximum (in expectation) accuracy $\mu^\star$. Thus, we do not focus on the probability of finding the best arm but aim to minimize the expected regret $\mathbb{E}[r_T]$. This regret should tend to 0, meaning that with a larger budget T, the probability of selecting a (very) good model increases.

Remark 3.1. Stronger results are generally of the type "$\mathbb{P}(r_T > \varepsilon) < \delta$", see, e.g., Zhao et al. [2023].

In order to obtain an upper bound for the expected regret, it is essential to make assumptions about the distribution of the accuracy expectations.

Assumption 3.2. When a new untrained model f_k is sampled from the search space, the expectation of its accuracy μ_k is assumed to be an independent sample from the continuous uniform distribution on $[0, 1]$:

$$\mu_k \sim \mathcal{U}\big([0, 1]\big) \,. \tag{4}$$

This implies $\mu^\star = 1$.

Remark 3.3. In infinite-armed bandit framework (see, e.g., Wang et al., 2008), classical assumptions generally characterize the probability of pulling near-optimal arms: given $\mu^\star$ and $\beta > 0$ a parameter of the reward expectation distribution, the probability that a new arm is ε-optimal is of order ε^β: $\mathbb{P}(\mu_k \geq \mu^\star - \varepsilon) = \mathcal{O}(\varepsilon^\beta)$ for $\varepsilon \to 0$. This statement is satisfied for $\beta = 1$ as soon as Assumption 3.2 holds.

The validity of this hypothesis in our case study is discussed in Appendix B.1.

Theorem 3.4. *Fix a risk $\delta \in (0,1)$, a budget $T \geq 1$ and an initial number of untrained models $K < T$. Assume that Assumption 3.2 on the distribution of the reward expectations holds. Algorithm 1 running with an exploration parameter $E = \frac{25}{36}\frac{(T-K)}{K^3}\delta^2$ satisfies*

$$\mathbb{E}[\, r_T \,] \leq \frac{1}{K+1} \,+\, \delta \,+\, 2TK\mathrm{e}^{-\frac{T-K}{18K^3}\delta^2} \,. \tag{5}$$

The theorem is obtained by dividing the regret in two:

$$r_T = \mu^\star - \mu_{\widehat{I}_T} = \left(\mu^\star - \mu_{I^\star}\right) + \left(\mu_{I^\star} - \mu_{\widehat{I}_T}\right),$$

where the random variable $I^\star$ is the best arm among the K sampled ones. Using results on the distribution of the maximum of K independent variables sampled from the continuous uniform distribution on $[0,1]$, we get rid of the first term: its expectation equals $1/(K+1)$. The second one is also split in two. Under Assumption 3.2, the probability that the exploration parameter E is not well fitted for UCB-E is bounded by the risk δ. If, on the other hand, it is well fitted, the upper-bound proposed in Theorem 1 of Audibert et al. [2010] is satisfied since its assumptions hold. This leads to the last term in Equation (5). Details of the proof are provided in Appendix A. With a risk $\delta = 1/K$, we get

$$\mathbb{E}[\, r_T \,] \leq \frac{2}{K} + 2TK\mathrm{e}^{-\frac{T-K}{18K^5}} \,. \tag{6}$$

It remains to fix the number of initial arms to minimize this expectation. In particular, when the order of K is a power of budget T^α, taking $\alpha \in (0, 1/5)$ ensures that the regret bound is $\mathcal{O}(T^{-\alpha})$. Indeed, recalling that K must be an integer less than T, α has to be in $(0,1)$. Moreover, the second term of (6) converges to 0 as it is of order $T^{1+\alpha}\exp\left(-T^{1-5\alpha}\right)$. It then becomes negligible compared with the first term only if $\alpha < 1/5$. The regret bound is then of the order of the first term, namely $T^{-\alpha}$, with $\alpha \in (0, 1/5)$.

Remark 3.5. SiRI algorithm introduced by Carpentier and Valko [2015] and discussed in 2.1 which involves a more complex confidence bound, leads, under Assumption 3.2, to a better upper bound of order $\mathcal{O}(1/\sqrt{T})$.

Here, we have chosen to retain ∞-UCB-E, a vanilla version of UCB-E for infinite-armed bandits, as the entry point for the development of our algorithm presented below.

4 A UCB-based algorithm incorporating mutation operators from evolutionary algorithms

Mutant-UCB (see Algorithm 2) incorporates two main ideas into ∞-UCB-E. First of all, there is no point in multiplying the number of sub-trains for the same model: there generally comes a time when it is no longer useful, so we can potentially define a maximum number of sub-trains N. Note that this idea of a maximum quantity of resources that can be allocated to a single model was already present in the Hyperband algorithm (see Li et al., 2018). Furthermore, in model selection, it is not uncommon for similar models to perform similarly, which is why configuration selection methods may be so effective in this task. In general, the main problem lies in defining a distance between models: search spaces are usually high-dimensional and hyper-parameters are of various kinds (learning rate, type of activation function, number of neurons, etc.). While the notion of distance between two models is not easy to define, evolutionary algorithms offer a good compromise: they breed new individuals through crossover and mutation operations. Crossover operations mix two individuals, while mutation operations can be applied to a single individual in order to create "mutants", involving some tiny changes. Those "mutants" can be seen as neighbors of the initial point. We could therefore imagine that a model chosen by the algorithm could mutate to give rise to a new one, with the intuition that the mutant and its original model will have similar accuracies. To our knowledge, the inclusion of mutation operators in evolutionary algorithms in a bandit algorithm is completely new. These two ideas run completely counter to the assumption of the stochastic infinite-armed bandit framework. Conditionally on the chosen arm, rewards are not independent since we assume that successive sub-trains on a model improve its accuracy up to a certain point of convergence. Furthermore, when a new untrained model is created through a mutation operation performed on an original model, it is unlikely that their accuracy expectations are independent.

Like the ∞-UCB-E algorithm, Mutant-UCB starts with the first sub-train of K models. At each round $t = K+1, \ldots,$ it still chooses the next arm optimistically, by resolving Equation (3). For an arm k, we recall that N_k is the number of

Algorithm 2 Mutant-UCB

Inputs:
T budget
E exploration parameter
K initial number of models
N maximum number of sub-trains that can be allocated to a single model

Initialization
Sample K untrained models $f_1, \ldots, f_K$
For $k = 1, 2, \ldots, K$
Perform a first sub-train on f_k which becomes f_k^1
Get the reward $a_k = \mathrm{acc}\big(f_k^1, \mathcal{D}_{\mathrm{valid}}\big)$
Define $N_k = \overline{N}_k = 1$, $\hat{\mu}_k = a_k$

For $t = K+1, K+2, \ldots, (T-N+1)$
Choose $I_t \in \mathrm{argmax}_{k\in\{1,\ldots K\}} \left\{\hat{\mu}_k + \sqrt{\frac{E}{N_k}}\right\}$
Sample $X_t \sim \mathcal{B}(p_t)$ with $p_t = 1 - \overline{N}_{I_t}/N$
If $X_t = 1$:
Perform a sub-train: model $f_{I_t}^{\overline{N}_{I_t}}$ becomes $f_{I_t}^{\overline{N}_{I_t}+1}$
Get the reward $a_t = \mathrm{acc}\big(f_{I_t}^{\overline{N}_{I_t}+1}, \mathcal{D}_{\mathrm{valid}}\big)$
Update $\hat{\mu}_{I_t} = \frac{1}{\overline{N}_{I_t}+1}\big(r_t + \overline{N}_{I_t}\hat{\mu}_{I_t}\big)$, $N_{I_t} = N_{I_t} + 1$
and $\overline{N}_{I_t} = \overline{N}_{I_t} + 1$
Else :
Update the number of models $K = K + 1$
Create a mutant model f_K from $f_{I_t}^{\overline{N}_{I_t}}$
Perform a first sub-train on f_K which becomes f_K^1
Get the reward $a_t = \mathrm{acc}\big(f_K^1, \mathcal{D}_{\mathrm{valid}}\big)$
Define $N_K = \overline{N}_K = 1$, $\hat{\mu}_K = a_t$
Update $N_{I_t} = N_{I_t} + 1$

Finalization
Select the best model $\widehat{I}_T \in \mathrm{argmax}_{k\in\{1,\ldots K\}}\, \hat{\mu}_k$
Finalize its training by performing $N - \overline{N}_{\widehat{I}_T}$ sub-trains

Output: $f_{\widehat{I}_T}^N$

times the arm has been picked before round t - see Equation (2). We now introduce $\overline{N}_k$, the integer that counts the number of times the model associated with arm k has been trained. Once arm I_t is picked, with $p_t = 1 - \overline{N}_{I_t}/N$:

- a sub-train is performed on $f_{I_t}^{\overline{N}_{I_t}}$ with probability p_t or
- a mutation is performed on $f_{I_t}^{\overline{N}_{I_t}}$ with probability $1 - p_t$.

The mutation is performed on the trained model $f_{I_t}^{\overline{N}_{I_t}}$ - and not just f_{I_t} - to include the case where certain parameters of the model optimized during training (e.g., weights of neuron networks) are passed on to its mutant. We detail the mutation operation for our use-case in Section 5. When a mutation occurs, a new model is created, a first sub-train is performed and the model is added to the list of potential models to be retained at the end of the algorithm. Thus, the number of models K increases by 1 each time a mutant model is created.

Remark 4.1. When a new model comes into play, it is very likely to be quickly chosen by the algorithm, even if its accuracy is not good: the algorithm must explore this new possibility. The "sleeping bandit" framework, in which new arms may be added and/or become unavailable during the algorithm execution, is studied in Kleinberg et al. [2010]. It proposes a very natural extension of UCB: the Awake Upper Estimated Reward algorithm and shows there is no need to adapt the confidence bounds.

The probability p_t decreases as the model goes along its sub-trains and guarantees that it will not be trained more than N times. The more the model has been trained, the more likely it is to mutate when selected. The underlying idea is that

further training will probably have little effect or even overfit in the case of neural networks, and that if the algorithm selects this already well-trained model, it is because it may have good accuracy (and it will probably be the same for a mutant model). Note that the probability p_t is linear in N_{I_t}; this choice is arbitrary and we could quite easily have chosen another type of relationship, e.g., $p_t = 1 - \exp(N_{I_t} - N)$. The algorithm ends with a finalization phase: the best model $\widehat{I}_T$ is selected among the initial models and the mutant models and its training is completed with $N - \overline{N}_{\widehat{I}_T}$ additional sub-trains.

Remark 4.2. In order to obtain an upper bound on the regret, we think it is necessary to retain the stochastic bandit hypothesis on the distribution of rewards (conditionally on the chosen arms, rewards are independent, so the accuracy does not depend on the number of sub-trains performed) as well as that on the distribution of the expectations of the model accuracies in the initialization phase of the algorithm (see Assumption 3.2). But, in order to legitimize the idea of integrating mutation operations, and hopefully improve the bound, it seems essential to add an assumption about the distribution of the expectation of the accuracies of mutant models, e.g.,

$$\mathbb{E}\Big[\, \mu_k \,\Big|\, f_k \text{ is a mutant of } f_j^{\overline{N}_j} \,\Big] = \mu_j \,.$$

We could also model the correlations between the rewards associated with a model and those of its mutants (see, e.g, Gupta et al., 2021 which propose the correlated multi-armed bandits framework). Theses kind of assumption in our case study are discussed in Appendix B.2.

5 Experiments

In this section, we evaluate the performance of our algorithm Mutant-UCB, on neural networks optimization. In order to highlight the advantages of our method, we put ourselves in a case where we make no assumptions about the smoothness of the reward acc and we do not consider any distance between the elements f_k from our search space. We therefore compare our methods with three algorithms that are applicable in this case: a random search, the Hyperband algorithm and an evolutionary algorithm. This neural networks optimization is applied to three image classification data sets. Due to the difficulty of correctly setting the exploration parameter of the algorithm ∞-UCB-E (the theoretical value is generally too high), we do not compare ourselves with this algorithm. We also explain its poor performance by the impossibility of limiting the number of sub-trains for this algorithm. However, we discuss and compare ∞-UCB-E and Mutant-UCB on the toy MNIST data set (see Deng, 2012) in Appendix C.1.

5.1 Experiment design

Data sets. We performed our experiments using three image classification data sets, also used by Li et al. [2018] to introduce the Hyperband algorithm: CIFAR-10 (see Krizhevsky et al., 2009), Street View House Numbers (SVHN, see Netzer et al., 2011) and rotated MNIST with background images, also called MRBI (see Larochelle et al., 2007). These first two data sets contain 32×32 RGB images, while MRBI contains 28×28 gray-scale images. The labels for each data set are converted to integers between 0 and 9. We split each data set into a training, a validation and a testing set. The training set is used to optimize the model weights (namely to perform the sub-trains), while the validation set is used to evaluate the configurations in the selection model algorithms (i.e. to get the rewards). Finally, the accuracies of the configuration selected by the algorithms are computed on the testing set to assess their quality. CIFAR-10 has 35k image on the train set, 15k on the validation set and 10k on the test set, SVHN has 51k, 22k and 26k and MRBI 10k, 2k and 50k data points on the three sets respectively. For all data sets we standardized the images so the input has a mean of zero and a standard deviation of 1.

Search space: the pool of possible configurations. We used the DRAGON framework developped by Keisler et al. [2023] to encode our neural networks. Their article contains an explanation and a tutorial on how to use the associated implemented package. In this framework, neural networks are represented as directed acyclic graphs (DAGs), where the nodes represent the layers (e.g., recurrent, feed-forward, convolutional) and the edges represent the connections between them. The task on which we want to try our algorithms is image classification. To do so, we define a generic search space (the pool of possible configurations f_k) with DRAGON, dedicated to the task at hand. Any sampled configuration f_k will be made of two directed acyclic graphs. The first one processes 2D data, and can be made of 2D convolutions, 2D pooling, normalization and dropout layers. The second one consists in a flatten layer followed by MLPs (Multi-Layers Perceptrons) and normalization layers. A final MLP layer is added at the output of the model to convert the latent vector into the desired output format. The framework includes operators, namely mutations and crossovers to modify and thus optimize the graphs. The mutation operators modify the neural network architecture by adding, removing or modifying the nodes and the connections in the graph. They can also be applied within the nodes, on the neural network hyper-parameters (e.g., convolution layer kernel size or an activation function). Crossover involves exchanging parts of two graphs.

Table 1: Number of tested models and accuracies (in %) of the best model for random search (RS), asynchronous evolutionary algorithm (EA) and Mutant-UCB on CIFAR-10, MRBI and SVHN data sets.

Data set	CIFAR-10	MRBI	SVHN
RS	1 000 · 75.3	1 000 · 75.5	1 000 · 90.7
EA	1 000 · 77.1	1 000 · 79.5	1 000 · 91.9
Hyperband	2 400 · 75.4	2 400 · 75.9	2 400 · 91.0
Mutant-UCB	**3 399 · 79.5**	**3 463 · 80.5**	**3 471 · 92.4**

Sub-trains. We trained our neural networks using a cyclical learning rate, as proposed by Huang et al. [2017]. When the learning rate is low, the neural network reaches a local minimum. Right after the learning rate goes up again taking the model out of the local minimum. We consider in our experiments that a sub-train is one of this loop, with learning rate getting from its maximum to its minimum. We let N be the maximum number of sub-trains for a given element f_k from our search space.

Baselines. The random search, the Evolutionary Algorithm (EA), Hyperband and Mutant-UCB have all been implemented so that they can be used with the DRAGON framework. They all use the same training and validation functions to assess the neural networks performance, and share a common budget, namely T. For the random search, we randomly select $K_{\text{RF}} = T/N$ neural networks. For each of them we perform N sub-trains, resulting in T sub-trains in total. For the Evolutionary Algorithm, we implemented an asynchronous (or steady-state) version. Compared to the standard algorithm, the steady-state evolutionary algorithm enhances efficiency on High-Performance Computing (HPC) by producing two offsprings from the population as soon as a free process is available, rather than waiting for the entire population to be evaluated (see Liu et al., 2018). We set an initial population of size K_{EA}, where the deep neural networks are randomly initialized. We perform N sub-trains on each of these models. Then, we evolve the population using the mutation and crossover operators from the DRAGON framework. If a generated offspring is better than the worst model from the population, it replaces it. During the optimization procedure, we generate $T/N - K_{\text{EA}}$ offsprings and we perform N sub-trains on each, resulting in a total of T sub-trains. For Hyperband we ran the algorithm with its parameters R and η such that the total number of sub-trains is T and that each model can be trained only N times (see, Li et al., 2018 for further details). The algorithm Mutant-UCB starts with an initial population of K_{MUTANT} and runs with a budget of T. For a fair comparison, EA and Mutant-UCB use the same mutation operators. We set $K_{\text{EA}} << K_{\text{MUTANT}} \lesssim K_{\text{RF}}$. Indeed, as each configuration is fully trained in the evolutionary algorithm, K_{EA} must be much lower than T/N to allow the creation of a sufficient number of offsprings. Similarly, Mutant-UCB mutation operator will create new configurations during the optimization procedure. However the evolutionary algorithm will generate even more individuals with the crossover, so K_{MUTANT} may be higher than K_{EA}. We then set K_{MUTANT} a bit smaller than T/N. We emphasize that, with the creation of offsprings and mutants, the final number of evaluated models by the evolutionary algorithm and Mutant-UCB will be much higher than K_{EA} and K_{MUTANT}, respectively. In addition, the random search and the evolutionary algorithm fully train each configuration tested (of which there are T/N), while Hyperband and Mutant-UCB allow some of them to be partially trained (resulting in a final population of more than T/N configurations).

5.2 Results

We run the experiments with $T = 10\,000$, $N = 10$ and $E = 0.05$ for Mutant-UCB. The tuning of the parameter E for Mutant-UCB is not as important as for the ∞-UCB-E algorithm. We detail this discussion in Appendix C.1. We only need the algorithm to not explore too much, since the mutation operator represents an additional form of exploration. Each sub-trains contains 10 epochs, resulting in a maximum of 100 training epochs, and the learning rate is set to 0.01. Each experiment is run on a HPC environment using 20 NVIDIA V100 GPUs.

We display Table 1 the maximum accuracies and the number of tested models for each algorithm from our baseline. We see that Mutant-UCB outperforms the random search, the evolutionary algorithm and Hyperband for every data sets. The use of the mutation operator seems to be the primary factor in this performance. Indeed, the evolutionary algorithm comes well ahead of Hyperband and the random search. It would seem that digging around promising solutions leads to better configurations. However, resources allocation also seems to be a key factor. Hyperband is in fact slightly better than the random search, and converges much faster, as it can be seen in Figure 1. The computation times to perform the T iterations vary a lot between the algorithms and the task at hand. The hardware definitely has an impact on this, but the resources allocation will also play a crucial role. Throughout the paper there is an implicit assumption that a sub-train's budget in terms of memory and time is independent of the model, which is not entirely correct. Indeed,

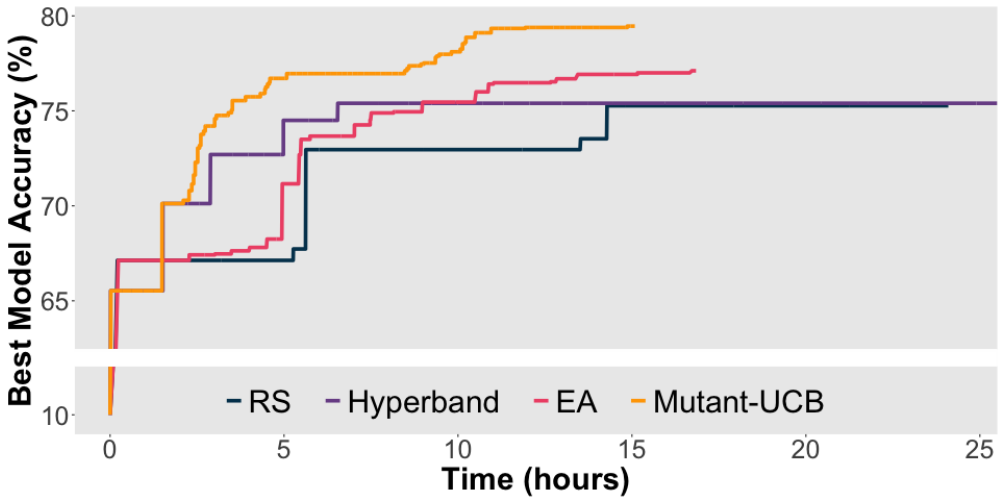

Figure (a): CIFAR-10.

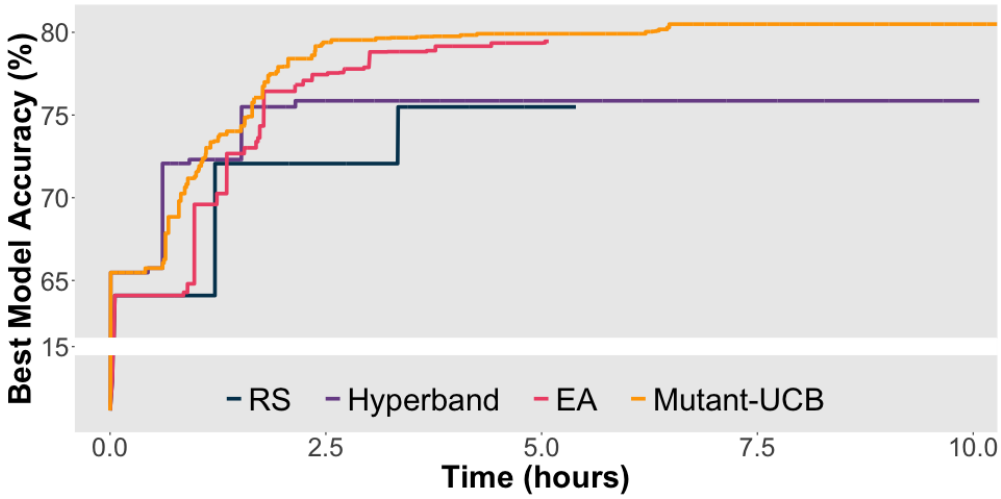

Figure (b): MRBI.

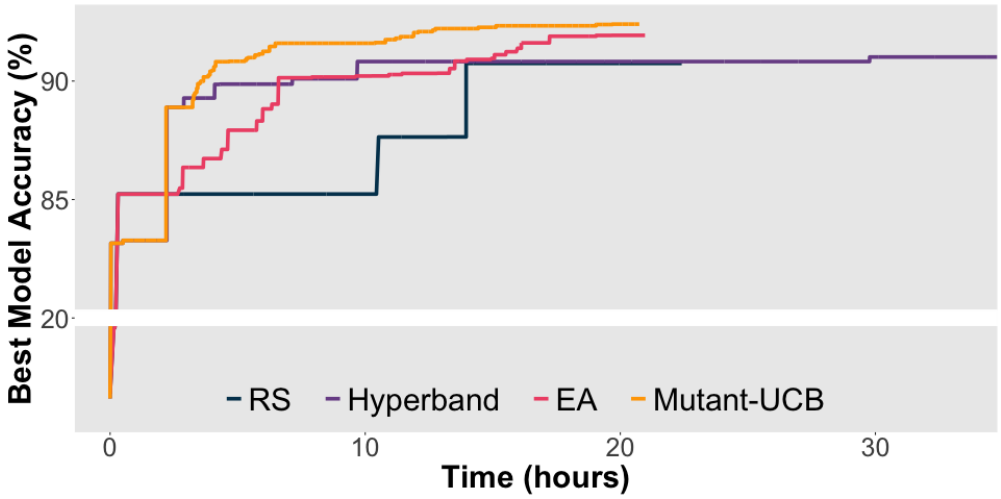

Figure (c): SVHN.

Figure 1: Accuracy of the best model over computational time for random search (RS), asynchronous evolutionary algorithm (EA) and Mutant-UCB on CIFAR-10, MRBI and SVHN data sets.

performing more sub-trains on configurations which are more complex take generally a longer time and affect the total duration of the experiment. Mutant-UCB, with both the mutation operator and the resources allocation has the fastest convergence and yields the better accuracies. Appendix C.2 details the models found by the different algorithms.

Codes are available in the supplementary material.

6 Discussion

This article presents Mutant-UCB, an innovative model selection algorithm, which combines a UCB-based bandit algorithm with evolutionary algorithm operators. Most configuration selection approaches, such as Bayesian optimization or continuous bandit algorithms, typically consider a normed vector space to represent the pool of possible configurations. These approaches assume that the reward function is smooth, meaning that two configurations that are close in the underlying vector space lead to close accuracies. Mutant-UCB and the other algorithms in the baseline do not require any smoothness assumptions. Besides, thanks to its resource allocation, Mutant-UCB demonstrates a high exploratory potential. It can evaluate more models within a similar budget compared to random search or evolutionary algorithms. For example, on the MRBI data set, with a budget $T = 10000$, Mutant-UCB evaluated 3500 configurations, while the Evolutionary Algorithm and the random search only evaluated 1000. The use of a mutation operator, on the other hand, reinforces the exploitation of promising solutions and allows us to reach much higher performance configurations than Hyperband and the random search. The mutation can be viewed as a concept of proximity: a mutant and its original model are close together, as defined by the chosen operator (which does not require any normed vector space). It remains more permissive than the operators of the evolutionary algorithm. In particular, the crossover from the evolutionary algorithm requires homogeneity between the elements of the search space, unlike Mutant-UCB. Thus, Mutant-UCB could be used with a search space that combines various machine learning models, such as neural networks, random forests, or boosting; as soon as we define a mutation operator for each type of model. Finally, the Mutant-UCB algorithm is highly scalable in an HPC environment because configurations are evaluated independently and asynchronously, in contrast to Hyperband and classical evolutionary algorithms, which evaluate populations synchronously. In summary, Mutant-UCB has several advantages that make it an attractive algorithm, in addition to its baseline-beating performance demonstrated in the previous section. One disadvantage is the need to store the weights of all previous configurations evaluated. This is because the pool of solutions on which we apply the UCB part of the algorithm is not limited, unlike the other algorithms from the baseline.

Prospects. The experiments demonstrate the relevance of Mutant-UCB. As mentioned in Remark 4.1, a challenge for further work would be to obtain an upper bound on the regret of this algorithm. To do so, we believe that it will be necessary to introduce some concepts from sleeping bandits due to the creation of mutants (see Kleinberg et al., 2010) and contextual bandits to model the proximity between mutants and their original models (see, among other Li et al., 2010). It should also be noted that the HPC environment and the neural network training duration puts us in a context where rewards arrive with a delay (namely, in a delayed bandits framework - see, e.g., Vernade et al., 2020) and not all arms are available at all times (sleeping bandits again).

In this paper we applied Mutant-UCB to a very generic problem: neural networks optimization for image classification. The flexibility of this algorithm means that it can be applied to a wide range of problems. A natural extension of this paper would be to apply Mutant-UCB to a variety of tasks, models and search spaces where state-of-the-art algorithms, by their very nature, would be limited or even unusable.

A Proof of Theorem 3.4

Among the arms sampled by the algorithm, we denote by $I^\star$ the best one, *i.e* $I^\star \in \operatorname{argmax}_{k\in\{1,\dots,K\}} \mu_k$. In order to minimize the regret, we need to draw enough arms to have a chance of finding one close enough to $\mu^\star$ but we also need to keep K small enough so that the UCB-E algorithm has a chance of converging on the best arm among those drawn (namely $\widehat{I}_T = I^\star$). Therefore, we decompose the regret this way:

$$\begin{aligned} r_T &= \mu^\star - \mu_{\widehat{I}_T} \\ &= (\mu^\star - \mu_{I^\star}) + (\mu_{I^\star} - \mu_{\widehat{I}_T})\,. \end{aligned}$$

By rearranging the arm reward expectations in descending order: $1 = \mu^\star \geq \mu_{(1)} \geq \cdots \geq \mu_{(K)} \geq 0$ and recalling that the k^{th} order statistics sampled from a uniform distribution is a beta-distributed random variable: $\mu_{(k)} \sim \mathrm{Beta}(K+1-k, k)$ - see e.g, David and Nagaraja [2004] - we get that $\mathbb{E}[\,\mu_{I^\star}] = \mathbb{E}[\,\mu_{(1)}] = K/(K+1)$. Using the linearity for the expectation and recalling that $\mu^\star = 1$, we get that $\mathbb{E}\,[\,\mu^\star - \mu_{I^\star}] = 1/(K+1)$. It remains to deal with $\mathbb{E}\,[\,\mu_{I^\star} - \mu_{\widehat{I}_T}]$ to conclude the proof.

The probability of the UCB-E algorithm succeeding depends on the difficulty of the underlying bandit problem. In Theorem 1 of Audibert et al. [2010], this "difficulty" is assumed to be known in advance and the exploration parameter is set accordingly. Here, we recall that the best arm is a random variable drawn at the start of the ∞-UCB-E algorithm. To rid of the expectation $\mathbb{E}\,[\,\mu_{I^\star} - \mu_{\widehat{I}_T}]$, we have to condition it on the event "Theorem 1 of Audibert et al. [2010] holds". Once the K arms have been sampled and $\mu_{(1)}, \dots \mu_{(K)}$ are set, the statement of the theorem is as follows: UCB-E satisfies

$$\mathbb{P}(\widehat{I}_T \leq I^\star) \leq 2TK\mathrm{e}^{-\frac{2E}{25}} \quad \text{if} \quad 0 \leq E \leq \frac{25}{36}\frac{T-K}{H_1} \quad \text{with} \quad H_1 = \sum_{k=1}^{K-1} \frac{1}{\Delta_{(k)}^2} \quad \text{and} \quad \Delta_{(k)} = \mu_{(1)} - \mu_{(k+1)}\,.$$

By noticing that $H_1 \leq K/\Delta_{(1)}^2$, a condition on $\Delta_{(1)}$ will be enough. Thus, using the law of total expectation with a conditioning on the sampled arms (on $\mu_{(1)}, \dots \mu_{(K)}$, namely), for any $h \in [0,1]$, we have:

$$\begin{aligned} \mathbb{E}[\,r_T\,] &= \mathbb{E}\Big[\mathbb{E}\big[\mu^\star - \mu_{I^\star} + \mu_{I^\star} - \mu_{\widehat{I}_T} |\, \mu_{(1)}, \dots \mu_{(K)}\big]\Big] = \mathbb{E}\big[\mu^\star - \mu_{I^\star}\big] + \mathbb{E}\Big[\,\mathbb{E}\big[\mu_{I^\star} - \mu_{\widehat{I}_T} | \Delta_{(1)}\big]\Big] \\ &= \frac{1}{K+1} + \mathbb{P}(\Delta_{(1)} \leq h)\mathbb{E}\big[\underbrace{\mu_{I^\star} - \mu_{\widehat{I}_T}}_{\leq 1} | \Delta_{(1)\leq h}\big] + \underbrace{\mathbb{P}(\Delta_{(1)} > h)}_{\leq 1}\mathbb{E}\big[\mu_{I^\star} - \mu_{\widehat{I}_T} | \Delta_{(1)>h}\big] \\ &\leq \frac{1}{K+1} + \mathbb{P}(\Delta_{(1)} \leq h) + \mathbb{P}\big(I^\star \neq \widehat{I}_T \,|\, \Delta_{(1)} > h\big)\,. \end{aligned} \tag{7}$$

The second term in Equation (7) can be bounded by Kh thanks to Lemma A.1 stated next. Proof of this is provided at the end of this appendix.

Lemma A.1. *Assume that Assumption 3.2 holds. When $K \geq 2$ arms are picked, with $\Delta_{(1)} := \mu_{(1)} - \mu_{(2)}$ the difference of the expectations of the two best arms, we have, for all $h \in [0,1]$,*

$$\mathbb{P}\big(\Delta_{(1)} \leq h\big) = 1 - (1-h)^K \leq Kh\,.$$

By choosing $h = \delta/K$, we get:

$$\mathbb{P}(\Delta_{(1)} \leq h) \leq \delta\,.$$

With $E = \frac{25}{36}\frac{(T-K)\delta^2}{K^3}$ (the exploration parameter set in Theorem 3.4), we get $E = \frac{25}{36}\frac{(T-K)h^2}{K}$. Conditionally to $\Delta_{(1)} \leq h$, as $H_1 \leq K/\Delta_{(1)}^2$, we have $E \leq \frac{25}{36}\frac{T-K}{H_1}$, i.e. Theorem 1 of Audibert et al. [2010] holds. It is used to bound the third term in Equation (7):

$$\mathbb{P}\big(I^\star \neq \widehat{I}_T \,|\, \Delta_{(1)} > h\big) \leq 2TKe^{-\frac{T-K}{18K}h^2}\,.$$

The proof is concluded by collecting all pieces.

Proof. Assumption 3.2 ensures that the expectations of the arms are independent and sampled from a uniform distribution: $\mu_1, \dots, \mu_K \overset{\text{i.i.d}}{\sim} \mathcal{U}([0,1])$. In Example 2.3. of the book *Order Statistics*, David and Nagaraja show that the order statistics $\mu_{(1)} \geq \cdots \geq \mu_{(K)}$ satisfy, for any $1 \leq i \leq j \leq K$,

$$\mu_{(i)} - \mu_{(j)} \sim \mathrm{Beta}(j-i, K+1-j+i)\,.$$

Therefore, with $i = 1$ and $j = 2$, we get $\Delta_{(1)} \sim \text{Beta}(1, K)$ and, for any $h \in [0, 1]$),

$$\mathbb{P}(\Delta_{(1)} \leq h) = \frac{1!\,K!}{(K-1)!} \int_0^h x^{1-1}(1-x)^{K-1} \mathrm{d}x = \int_{1-h}^1 K t^{K-1} \mathrm{d}t = \left[t^K \right]_{1-h}^1 = 1 - (1-h)^K \,.$$

To conclude the proof, it remains to show that $1-(1-h)^K \leq Kh$. By studying the function $f : h \mapsto Kh-1+(1-h)^K$ on $[0, 1]$ and with $K \geq 2$, it is straightforward:

h	0		1
$f'(h) = K - K(1-h)^{K-1}$	0	$+$	K
$f(h) = Kh - 1 + (1-h)^K$	0	↗	1

□

B Models accuracy distributions

B.1 Accuracy distributions for CIFAR-10, MRBI and SVHN data set

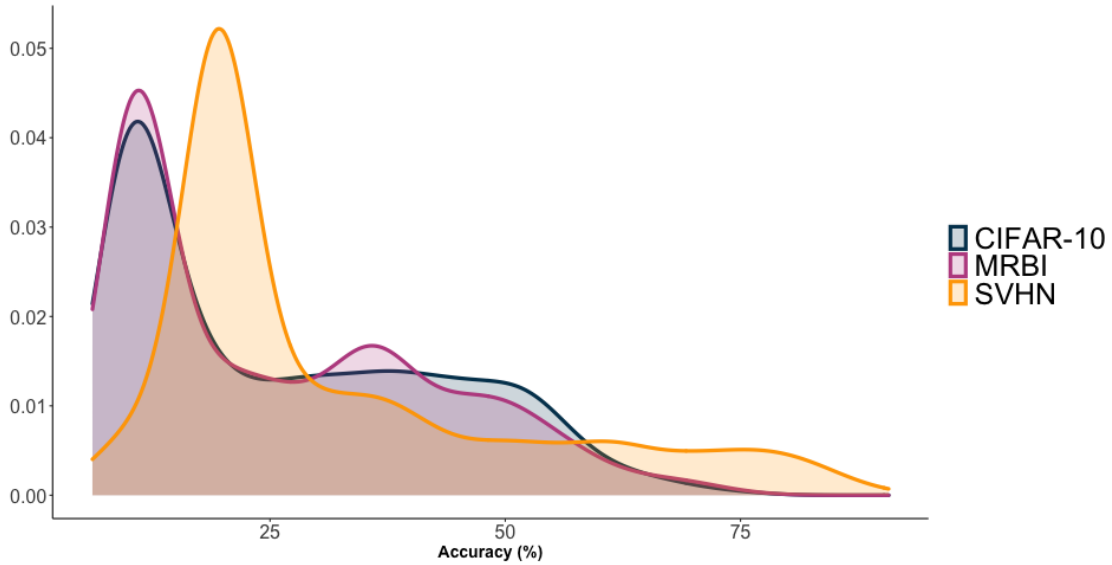

Figure 2: Distributions of the accuracy of 2 000 models randomly sampled and trained for the classification of CIFAR-10, MRBI and SVHN.

Figure 2 shows the density of accuracies obtained for 2 000 models randomly sampled in the search space and fully trained for the classification task, on the CIFAR-10, MRBI and SVHN data sets. It should be noted that since the seed is fixed, the three data sets share the same 2 000 models in terms of architectures and hyper-parameters. However, their weights are obviously different due to the training on the specific data set. It is quite clear that Assumption 3.2 does not hold. As mentioned in Remark 3.3, the polynomial dependency is important for the tail of the distribution (ε^β: $\mathbb{P}(\mu_k \geq \mu^\star - \varepsilon) = \mathcal{O}(\varepsilon^\beta)$ for $\varepsilon \to 0$). It is not very clear here; you should have to zoom in and train an unmanageable number of models.

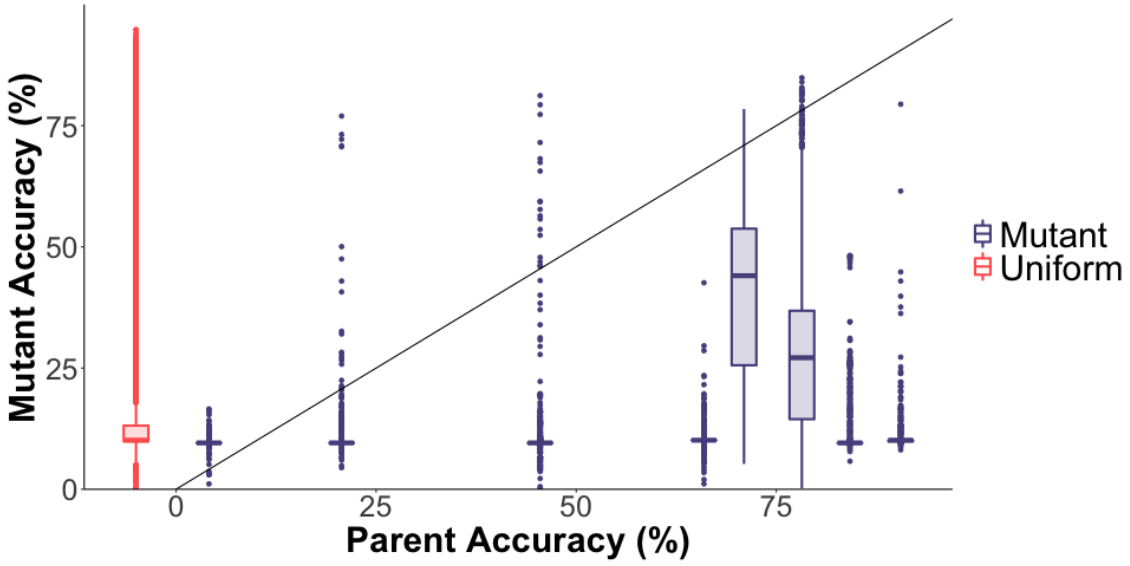

Figure 3: Mutants distributions for the MNIST data set. For eight configurations with various accuracies we trained and evaluated $4\,000$ mutants. The boxplots in purple represent the accuracies distribution for each configurations, ordered by their accuracy. The boxplot in pink represents the random search.

B.2 Mutants accuracy distributions

We consider here height configurations, trained on the toy MNIST data set (see Deng, 2012), of various accuracies. For each model, we generate and train $4\,000$ mutants. In Figure 3, we plot in purple the accuracies boxplots of the mutants from this height configurations. These boxplots are ordered according to the accuracy of the parent model. The boxplot in pink represents the accuracies of the random search on MNIST. In black, we plot the function $x \mapsto x$. Empirical means of mutant accuracies do not lie on the first bisector (i.e the black line). This means the equation

$$\mathbb{E}\Big[\, \mu_k \,\Big|\, f_k \text{ is a mutant of } f_j^{\overline{N}_j} \,\Big] = \mu_j \,,$$

does not hold. It is furthermore difficult to come up with any reasonable assumption concerning a potential link between the accuracies of a parent and its mutants. This question requires further study using different data sets and mutation operators. However, this can be costly from a computational perspective.

C Additional information on the experiments

C.1 Discussion on the exploration parameter tuning

Table 2: Number of tested models and accuracy (in %) of the best model for ∞-UCB-E and Mutant-UCB run with various exploration parameters.

Algorithm	∞-UCB-E	Mutant-UCB
$E = 0.01$	$3\,000 \cdot 95.9$	**$2\,853 \cdot 98.5$**
$E = 0.05$	$3\,000 \cdot 97.2$	**$2\,828 \cdot 98.7$**
$E = 1$	**$3\,000 \cdot 97.2$**	**$2\,629 \cdot 97.2$**
$E = 5$	**$3\,000 \cdot 97.2$**	$2\,393 \cdot 96.7$
$E = 10$	$3\,000 \cdot 95.9$	**$2\,435 \cdot 97.6$**

We discuss here the tuning of the exploration parameter E of our algorithms. We run experiments on the toy MNIST data set (see Deng, 2012) with $T = 10\,000$, $N = 10$ and various values for E. Results are in Table 2. They attest the performance of Mutant-UCB compare to our first algorithm ∞-UCB-E. Figure 4 illustrates the difficulty to find for this second algorithm the exploration parameter E that correctly balances exploration and exploitation. With $T = 10\,000$,

to satisfy the regret bound proved in Theorem 3.4, the exploration parameter E should be between $E = 0.00065$ (α close to 1) and $E = 1$ ($\alpha = 1/5$, the best theoretical value). As the theory is generally very conservative, we consider values between 0.01 and 10. For $E = 10$, the algorithm ∞-UCB-E explores too much, which leads it to sub-train each model a little (it only under-train its best model 6 times). Conversely, for $E = 0.01$, the algorithm exploits too much and concentrates on a few models, unfortunately not the best (406 sub-trains are performed on its best model). For Mutant-UCB, Figure 5 shows that the exploration parameter tuning seems less crucial. Thanks to the limitation in the sub-train allocations, it is more difficult to over-exploit, since in this case more and more mutants are created. On the other hand, over-exploration is possible: as with ∞-UCB-E, if E is too large, few models get a lot of sub-trains. Note once again the interest of mutations: almost every time, Mutant-UCB found a model that outperforms the best model in the initial population of ∞-UCB-E (containing $3\,000$ models),

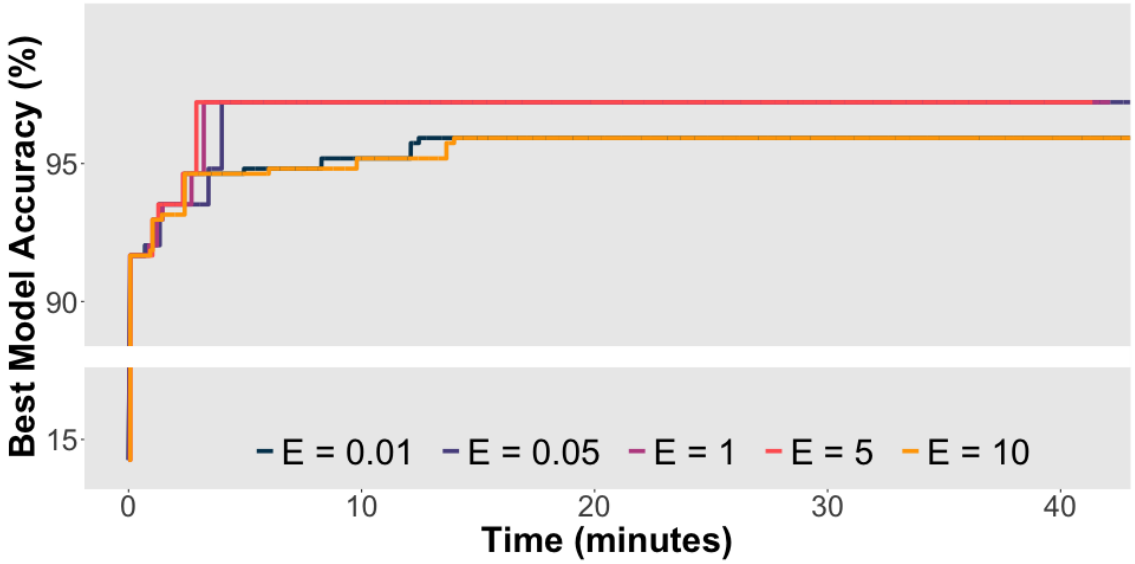

Figure 4: Maximum accuracy over computation time for the algorithm ∞-UCB-E ran with various exploration parameters ($E = 0.01$, $E = 0.05$, $E = 1$, $E = 5$ and $E = 10$) and a population of $K = 3\,000$ on MNIST data set.

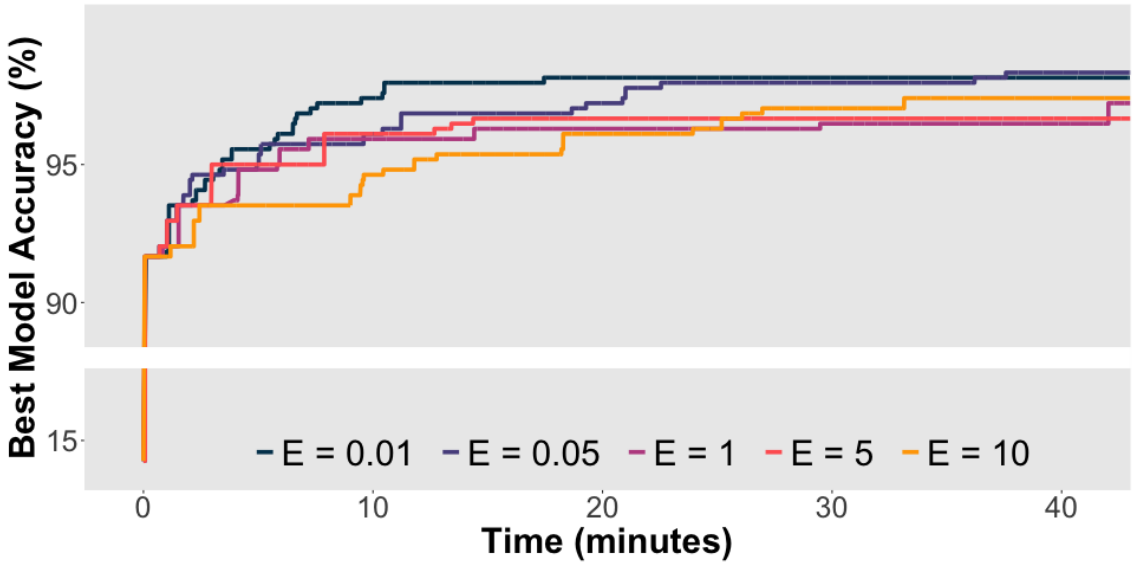

Figure 5: Maximum accuracy over computation time for the algorithm Mutant-UCB ran with various exploration parameters ($E = 0.01$, $E = 0.05$, $E = 1$, $E = 5$ and $E = 10$) and an initial population of $K = 500$ on MNIST data set.

C.2 Models found by the algorithms

We set the same general seed for each algorithm and data set: CIFAR-10, MRBI and SHVN. This means that the initial pool of solutions is exactly the same for each algorithm and for the three data sets. Hyperband and the random search found exactly the same architecture, indicating that the best configuration was at the beginning of this pool. Indeed, despite the fact that Hyperband looks at more than twice as many configurations as the random search, it resulted in the same configuration as the random search. Hyperband's best results come from the fact that 10 sub-trains are probably too many for this model, which overfits in the case of random search. More surprisingly, the best model for Hyperband and random search, shown in Figure 6, is identical for all three data sets.

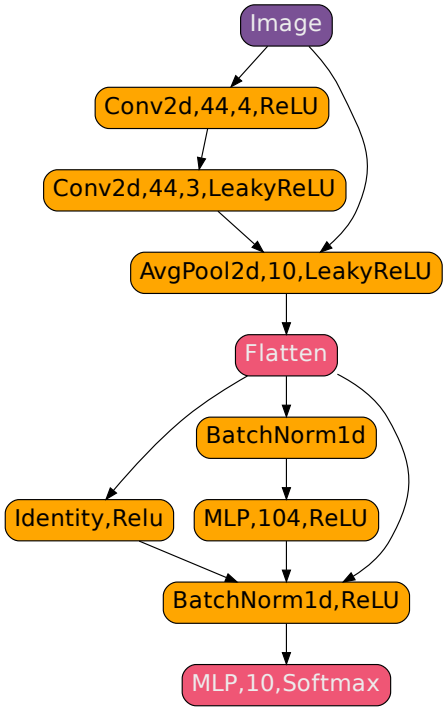

Figure 6: Best model found for CIFAR-10, MRBI and SVHN by the random search and Hyperband

Thanks to their evolutionary tools, the evolutionary algorithm and Mutant-UCB where able to create more performing configurations. Figure 7 shows the models found by both algorithms on the CIFAR-10 data set. The one found by Mutant-UCB, in Figure (b), is really close to the original one from the pool displayed Figure 6. The mutation operator was used to add more MLP layers at the end of the neural network which helps improving the model accuracy. On the other hand, the structure shown Figure (a), found by the evolutionary algorithm is very different. In this algorithm, a new configuration is created by crossing two parents with the crossover and then applying a mutation to one of the offspring. This double transformation allows to move much further away from the initial pool of solutions. In the case of the CIFAR-10 data set, this led the algorithm to consider very complex structures, which was not necessary to obtain good performance.

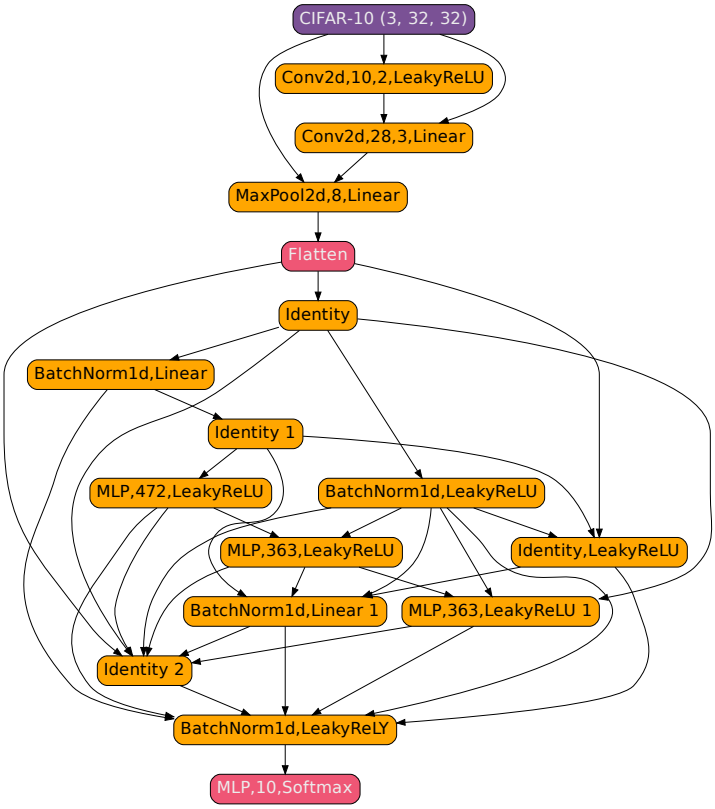

Figure (a): Evolutionary algorithm.

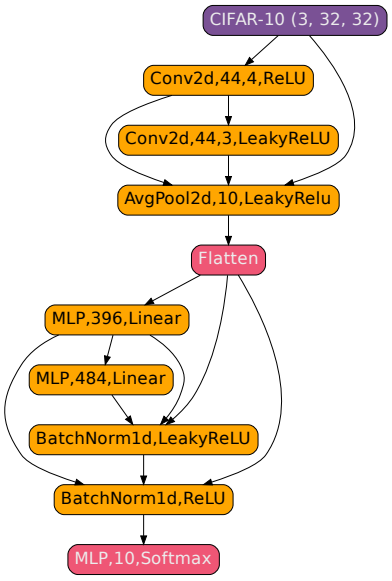

Figure (b): Mutant-UCB.

Figure 7: Best configurations found by the evolutionary algorithm and Mutant-UCB on the CIFAR-10 data set.

References

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

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

Noor Awad, Neeratyoy Mallik, and Frank Hutter. Dehb: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization, 2021.

Maryam Aziz, Jesse Anderton, Emilie Kaufmann, and Javed Aslam. Pure exploration in infinitely-armed bandit models with fixed-confidence. In *Algorithmic Learning Theory*, pages 3–24. PMLR, 2018.

Donald A Berry, Robert W Chen, Alan Zame, David C Heath, and Larry A Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, 25(5):2103–2116, 1997.

Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *International Conference on Machine Learning*, pages 1133–1141. PMLR, 2015.

Zhongxiang Dai, Gregory Kang Ruey Lau, Arun Verma, Yao Shu, Bryan Kian Hsiang Low, and Patrick Jaillet. Quantum bayesian optimization. *arXiv preprint arXiv:2310.05373*, 2023.

Herbert A David and Haikady N Nagaraja. *Order statistics*. John Wiley & Sons, 2004.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer, 2015.

Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35, 2020.

Samarth Gupta, Shreyas Chaudhari, Gauri Joshi, and Osman Yağan. Multi-armed bandits with correlated arms. *IEEE Transactions on Information Theory*, 67(10):6711–6732, 2021.

Matthew Hoffman, Bobak Shahriari, and Nando Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Artificial Intelligence and Statistics*, pages 365–374. PMLR, 2014.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Hanxun Huang, Xingjun Ma, Sarah M Erfani, and James Bailey. Neural architecture search via combinatorial multi-armed bandit. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

Zheng Jian, Han Wenran, Zhang Ying, and Ji Shufan. Eenas: An efficient evolutionary algorithm for neural architecture search. In *Asian Conference on Machine Learning*, pages 1261–1276. PMLR, 2023.

Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. *Advances in neural information processing systems*, 29, 2016.

Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. *Advances in neural information processing systems*, 31, 2018.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *International conference on machine learning*, pages 1238–1246. PMLR, 2013.

Julie Keisler, El-Ghazali Talbi, Sandra Claudel, and Gilles Cabriel. An algorithmic framework for the optimization of deep neural networks architectures and hyperparameters. *arXiv preprint arXiv:2303.12797*, 2023.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2-3):245–272, 2010.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480, 2007.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Ke Li, Alvaro Fialho, Sam Kwong, and Qingfu Zhang. Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 18(1):114–130, 2013.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search, 2018.

Gustavo Malkomes, Charles Schaff, and Roman Garnett. Bayesian optimization for automated model selection. *Advances in neural information processing systems*, 29, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Aditya Rawal and Risto Miikkulainen. From nodes to networks: Evolving recurrent neural networks, 2018.

Xuedong Shang, Emilie Kaufmann, and Michal Valko. A simple dynamic bandit algorithm for hyper-parameter tuning. 2019.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Ivana Strumberger, Eva Tuba, Nebojsa Bacanin, Raka Jovanovic, and Milan Tuba. Convolutional neural network architecture design by the tree growth algorithm framework. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

Kei Terayama, Masato Sumita, Ryo Tamura, and Koji Tsuda. Black-box optimization for automated discovery. *Accounts of Chemical Research*, 54(6):1334–1346, 2021.

Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brueckner. Linear bandits with stochastic delayed feedback. In *International Conference on Machine Learning*, pages 9712–9721. PMLR, 2020.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. *Advances in Neural Information Processing Systems*, 21, 2008.

Steven R Young, Derek C Rose, Thomas P Karnowski, Seung-Hwan Lim, and Robert M Patton. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In *Proceedings of the workshop on machine learning in high-performance computing environments*, pages 1–5, 2015.

Yao Zhao, Connor Stephens, Csaba Szepesvári, and Kwang-Sung Jun. Revisiting simple regret: Fast rates for returning a good arm. In *International Conference on Machine Learning*, pages 42110–42158. PMLR, 2023.

Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2016.