LUCIE: An Evaluation and Selection Method for Stochastic Problems

Erwan Lecarpentier*
University of Toulouse, IRIT (UMR 5505), REVA team
IRT Saint Éxupery
Toulouse, France
erwanlecarpentier@mailbox.org

Paul Templier*
University of Toulouse
ISAE-SUPAERO
Toulouse, France
paul.templier@isae-supaero.fr

Emmanuel Rachelson
University of Toulouse
ISAE-SUPAERO
Toulouse, France
emmanuel.rachelson@isae-supaero.fr

Dennis G. Wilson
University of Toulouse
ISAE-SUPAERO
Toulouse, France
dennis.wilson@isae-supaero.fr

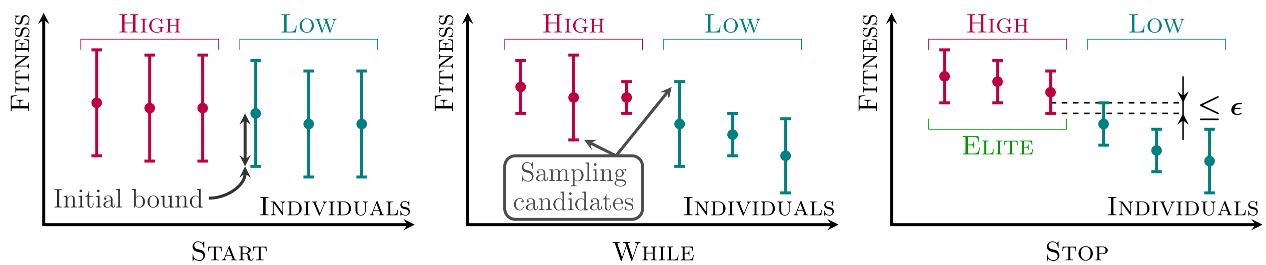

Figure 1: The LUCIE individuals selection procedure.

ABSTRACT

Selection in genetic algorithms is difficult for stochastic problems due to noise in the fitness space. Common methods to deal with this fitness noise include sampling multiple fitness values, which can be expensive. We propose LUCIE, the Lower Upper Confidence Intervals Elitism method, which selects individuals based on confidence. By focusing evaluation on separating promising individuals from others, we demonstrate that LUCIE can be effectively used as an elitism mechanism in genetic algorithms. We provide a theoretical analysis on the convergence of LUCIE and demonstrate its ability to select fit individuals across multiple types of noise on the OneMax and LeadingOnes problems. We also evaluate LUCIE as a selection method for neuroevolution on control policies with stochastic fitness values.

CCS CONCEPTS

• **Computing methodologies → Search methodologies**.

*Both authors contributed equally to this research.

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528838

KEYWORDS

Evolutionary Algorithm, Stochastic Fitness, Confidence Bounds

ACM Reference Format:
Erwan Lecarpentier, Paul Templier, Emmanuel Rachelson, and Dennis G. Wilson. 2022. LUCIE: An Evaluation and Selection Method for Stochastic Problems. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528838

1 INTRODUCTION

Fitness evaluation of individuals in practical Evolutionary Algorithms (EAs) applications is often subject to noise, and estimating the true individual expected fitness across the population can easily become costly. In this context, identifying the best candidates for elitism with as few fitness samples as possible is an appealing feature. We consider the problem of optimizing a fitness function when only noisy observations of this fitness at queried solution points are available to an algorithm. This case is very common, as *exact* evaluation of solutions may be prohibitively costly to obtain, impossible to simulate, or only possible with a non-exact simulation [7, 20]. In such a case, different evaluations of the same solution often feature high variance [27], in some cases intentionally injected into simulations to increase realism.

We approach this problem through improving the elitism mechanism in EAs. EAs follow a common pattern of 1) evaluating a population of solutions, called *individuals* or *candidates*, 2) generating a new population through selection and recombination of members of the former population, 3) repeating from step 1 [8]. Each completion of this loop is called a *generation*. An important step in many EAs is *elitism*, or the conservation of a subset of the best individuals from one generation to the next. Elitism is often used in genetic algorithms and multi-objective optimization to ensure that fit individuals remain in the population [3, 19]. The selection of these elites is usually done through taking the best individuals according to their fitness values, *i.e.*, truncation selection [30]. In the presence of noisy observations of this fitness, this selection phase can be misled and include individuals with low expected fitness value but high observed fitness. Misclassifying elites and non-elites may in turn prevent optimization in EAs.

Noisy fitness evaluations are often handled through fitness approximation, specifically through resampling and surrogate modelling methods. Resampling methods aggregate a fixed number of evaluations of an individual's fitness into an empirical mean, estimating the expected fitness of the individual. This method is simple but costly in terms of number of evaluations. In surrogate modelling methods, a model such as a Gaussian Process is constructed to approximate the fitness function. Surrogate modelling is often used for fitness approximation in EAs in general, as exact fitness values can be costly or difficult to calculate [22]. However, surrogate models are difficult to employ for certain EAs, for example neuroevolution where the dimensionality of the genotype may be very high or variable [13].

In this work, we propose LUCIE (Lower Upper Confidence Intervals Elitism), a method for efficient and accurate selection of elites. In LUCIE, this selection problem is cast as a Multi-Armed Bandit (MAB) problem of best μ arms identification. The arms, in this case, are the individuals of the population, as illustrated by Figure 1. For each individual, LUCIE maintains an empirical estimate of its expected fitness, along with a confidence interval. To efficiently identify elites, LUCIE sequentially samples individuals that are hard to classify as elite or non-elite. This sampling strategy allows the algorithm to perform selection with high confidence while running a limited number of evaluations. Furthermore, the use of confidence intervals allows to guarantee that the elite subset is correctly identified with high probability. Figure 1 illustrates this general process.

The outline of the paper is as follows. In Section 2, we introduce key notions and related work. Then, we describe the LUCIE algorithm in Section 3. In Section 4, we study LUCIE theoretically. We demonstrate that, during a generation, it selects ϵ-optimal elite members with probability at least $1 - \delta$ in a finite number of steps. We also formally demonstrate that LUCIE is able to converge to optimal solutions in the stochastic versions of the OneMax and LeadingOnes problems with only mild restrictions on the noise model [16]. In Section 5 we carry out an empirical analysis of LUCIE. First, we confirm the surprising ability of LUCIE to converge to optimal solutions of the OneMax and LeadingOnes problems, even under heavy levels of noise. We then apply LUCIE to neuroevolution of robotic control policies and observe again that LUCIE is able to find optimal policies even under heavy levels fitness noise. We conclude in Section 6.

2 BACKGROUND & RELATED WORK

Fitness approximation for EAs is a large domain which focuses on reducing computation cost of fitness evaluation, informing fitness evaluation with models or analysis of the objective function, or fully replacing evaluations with a surrogate model [21, 22]. Here, we focus on the specific case of overcoming noise in fitness evaluations. We first overview the problem of stochastic objective functions, then desribe the two families of approaches for this class of problems: resampling and surrogate models. Finally, we provide an overview of bandits algorithms, on which LUCIE is based.

2.1 Stochastic objective functions

Two main sources of stochasticity in fitness evaluation for EAs have been theoretically studied, namely, *prior* and *posterior* noise [26]. Prior noise [10, 16] refers to the stochastic perturbation of a solution prior to its evaluation through a deterministic fitness function. For example, in the noisy OneMax problem, a binary genotype is randomly modified by flipping a percentage of bits and then is evaluated with the standard OneMax function, i.e. the sum of genes. Posterior noise [11, 12, 16] refers to the addition of noise, a random variable sampled according to a defined distribution D, to the deterministic fitness evaluation of a solution. In [12], the difference between Gaussian and uniform noise posterior noise is studied for the OneMax problem, demonstrating that the $(\mu + 1) - EA$ scales to uniform but not Gaussian noise. We focus in this work on posterior noise.

In many applications, noise may be intrinsically linked to the fitness function. In robotics, for example, actuators may introduce prior noise by modifying actions randomly and sensors may give noisy state information [20]. In robot simulation environments, noise may be injected inside the simulation loop or during policy evaluation to encourage robust policies which can be deployed in the real world [27]; as such, there is both prior and posterior noise in robotics. Other applications which use simulation for optimization with EAs may contain similar sources of noise.

2.2 EAs for stochastic problems

Resampling. The first approach to overcoming noisy fitness functions is simply to collect more fitness samples, *i.e.*, resampling. Resampling methods aggregate a fixed number of evaluations of an individual's fitness into an empirical mean estimating the expected fitness of the individual. While this method is simple, it is costly in terms of evaluation time, while EAs often require many evaluations. Regarding classical resampling strategies, [1] propose a general resampling method to improve iterative optimization algorithms in stochastic settings. Their approach has been applied to EAs in [26], who find an upper-bound on the required number of samples per-individual to find an optimal solution with high probability. [9] suggest the use of empirical median rather than empirical mean while using resampling for selection. They demonstrate theoretically the advantage of the median in terms of robustness and sample complexity. [12] observe that EAs using myopic mutation operators essentially perform random walk in the stochastic setting. They propose a theoretical and empirical analysis of the Compact Genetic Algorithm (cGA), which learns the allele distribution of an optimal solution for a given problem, and show

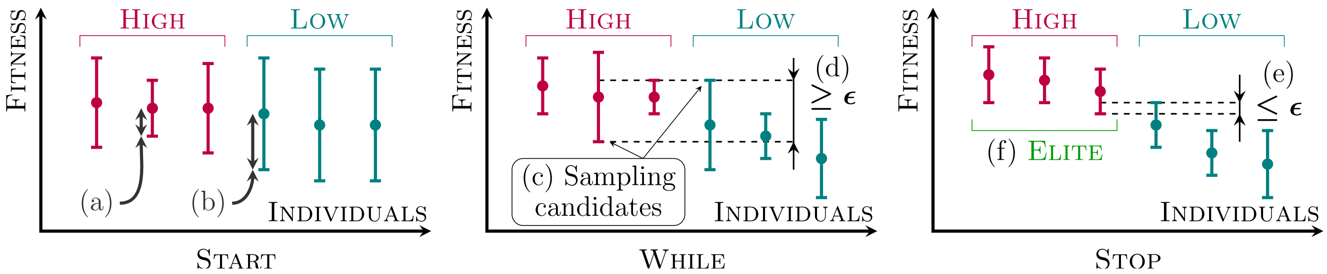

Figure 2: The LUCIE individuals selection procedure. (a) Individuals from previous generations may have already been sampled several times, inducing narrow confidence bounds. (b) New individuals are sampled once, inducing larger confidence bounds. (c) At each step, the two most ambiguous candidates regarding elitism are sampled. (d) If the uncertainty gap between the bounds of the sampling candidates is larger than ϵ, the process repeats itself. (e) It stops when the gap is lesser than ϵ. (f) The selected elite set is the μ individuals with highest estimated expected fitness.

better performance comparatively to resampling Random Local Search (RLS). Overall, the computational cost of resampling individuals fitness a fixed number of times hinders the applicability of resampling EA. LUCIE overcomes this aspect by selectively choosing individuals to evaluate, reducing the overall cost of selection.

Surrogate Modeling. The second approach for reducing the impact of noise in fitness evaluation is surrogate modelling. Those methods use a model to learn the fitness landscape, as a function of individual's features. [21, 22] survey surrogate-assisted EAs and show their benefits as both permitting fast evaluation of new individuals and driving exploration of the search space. The author also stress the limitations of surrogates which may mislead an optimizer to a false optima and be prohibitively costly (*e.g.*, Gaussian Processes [28, 31]). [21] also exhibit the limitation of modelling in that the expressiveness of the model should be able to capture the fitness landscape, which is sometimes a strong assumption. [31] use a Gaussian Process surrogate model to perform pre-selection of individuals during elite selection. The pre-selected subset is then refined by evaluating the most promising individuals. SAIL [13] improve the MAP-Elite algorithm [7] by modelling the relation between feature space and fitness function, yielding a better exploration-exploitation tradeoff. Similarly to SAIL, [18] propose to model the feature-objective mapping, but using hierarchical surrogates, improving scaling to the dimensionality of the feature space. [14] successfully apply surrogate modelling to neuroevolution. They leverage a distance measure between individuals to permit model predictions for neural networks. LUCIE removes the necessity of maintaining a supplementary and potentially costly model, as computational overhead of the method is negligible, which motivates the approach compared to surrogate modelling.

2.3 Best arms identification in MAB

The Multi-Armed Bandit is a problem of resource allocation: given a set of possible choices, or *arms*, each yielding a reward, one seeks to maximize one's expected gain as quickly as possible [25]. Besides the goal of gain maximization, bandits algorithms aiming at *identifying* the best arm or a set of best arms have been derived [5, 15, 24]. As our goal in this paper is to identify the best candidates within a population, we are interested in the latter. In this setting, [5] propose the Successive Accepts and Rejects (SAR) algorithm, identifying the best μ arms by sequentially accepting or rejecting individuals. [24] propose the Lower Upper Confidence Bounds (LUCB) algorithm for identifying the best μ arms with PAC guarantees, yielding an ϵ-close solution with probability at least $1 - \delta$, reached within a polynomial number of steps. They draw inspiration from the gain maximization UCB algorithm [2] to exploit confidence intervals in order to measure uncertainty and propose a provably efficient algorithm. We build on LUCB to propose the LUCIE selection method for EAs, presented in the next section.

3 THE LUCIE ALGORITHM

The LUCIE algorithm answers the problem of elite selection in evolutionary algorithms when fitness evaluation is subject to noise. Specifically, elitism consists in selecting the μ individuals with highest expected fitness, in a population of size $n = \mu + \lambda$. In noisy fitness contexts, poor elite selection methods cause good individuals to be lost between generations, which hinders the overall performance of the EA. Therefore, a good elite selection strategy predicts individual average fitnesses both *accurately* and *efficiently*, that is with as few samples as possible. In light of these two objectives, we cast this selection problem as a multi-armed bandit optimization problem where the arms are the individuals. The goal of the problem is then to identify the μ elites as quickly as possible with high accuracy. In the MAB literature, this formulation is close to the best μ arms identification problems tackled in [5, 23, 24]. We elaborate LUCIE on the foundation of the LUCB algorithm introduced by [24]. LUCB is a Probably Approximately Correct (PAC) [32] method for identifying the best μ arms in a polynomial number of steps with a controlled probability of failure. LUCIE extends the LUCB algorithm in the context of evolutionary algorithms, seeing each generation's elite selection problem as a best μ individuals identification problem. Therefore, LUCIE solves a sequence of separate, correlated bandit problems, by leveraging the history of fitness samples collected for elite individuals in previous generations.

3.1 Algorithm outline

During selection, LUCIE sequentially samples individual fitnesses to identify the μ elite members as fast as possible. Instead of uniformly sampling the individuals, it focuses on those that are hard to classify as belonging or not to the elite. To do so, for each individual, LUCIE maintains both an empirical estimate of its true fitness and a confidence bound on this estimate. At each step t of the current generation's selection process, the two "most ambiguous" individuals are sampled, and the confidence bounds of all individuals are updated. So at time step t of generation g, exactly $2t$ individual fitnesses have been drawn. Once the confidence bounds are tight enough to accurately distinguish elite members from the rest, the process terminates and the algorithm proceeds to the next generation. This process is illustrated on Figure 2 and Algorithm 1.

Algorithm 1: LUCIE selection algorithm

```
Sample each new individual at least once
Update bounds for all individuals
while Termination criterion of Equation (2) is not verified do
    Sample fitness of l^t and h^t (Equation (1))
    Update empirical fitnesses \hat{f}_{l^t}^{g,t} and \hat{f}_{l^t}^{g,t}
    Update bounds for all individuals
end
```

Consider generation g. We write Ind^g the set of individuals, *i.e.*, bandit arms, and f_i the true (unknown) expected fitness of $i \in \text{Ind}^g$. Without loss of generality and for the sake of simplicity, we shall assume in this section that $f_i \in [0, 1], \forall i$, and that the samples produced from the (stochastic) evaluation are also comprised between 0 and 1. For each individual, we maintain a memory of its fitness samples empirical average, and the number of times it has been evaluated since it was spawned. We write $u_i^{g,t}$ the total number of fitness samples collected over the lifespan of individual i, until step t of generation g. Note that this includes samples collected in previous generations. The true expected fitness of each individual is estimated based on the empirical mean of these samples, denoted by $\hat{f}_i^{g,t}$. We write Top^g the set of the μ individuals with highest (true) expected fitness, and, Bot^g the λ remaining individuals. For $\epsilon \in [0, 1]$, we define Good^g the set of (ϵ, μ)-optimal individuals, and Bad^g its complement in Ind^g, as

$$\text{Good}^g \stackrel{\text{def}}{=} \{i \in \text{Ind}^g,\ f_i \geq \min_{j \in \text{Top}^g} f_j - \epsilon\},$$

$$\text{Bad}^g \stackrel{\text{def}}{=} \text{Ind}^g \setminus \text{Good}^g.$$

As is, Good^g contains candidates that are either in Top^g or at most ϵ worse than Top^g's weakest member. The objective of LUCIE is to return a set of μ individuals in Good^g at each generation. At each step, the algorithm separates the population in two sets: $\text{High}^{g,t}$, the set of the μ individuals with largest fitness estimates, and $\text{Low}^{g,t}$, the set of the λ remaining individuals. $\text{High}^{g,t}$ is the set of current best elite estimates. Appendix 1 provides a graphical summary of the sets defined herein.

The **confidence intervals** are used to determine both the sampling strategy and the termination criterion. Intuitively, they are defined so that an individual's true expected fitness falls with high probability between its lower and upper confidence bounds. We makes use of a bounding function $\beta : \mathbb{N}^2 \to \mathbb{R}$. The upper confidence bound of an individual i is defined as $\hat{f}_i^{g,t} + \beta\left(u_i^{g,t}, t\right)$, and its lower confidence bound as $\hat{f}_i^{g,t} - \beta\left(u_i^{g,t}, t\right)$. Before giving a closed-form expression of β in Section 4, we explain how it is employed in the sampling strategy and the stopping criterion of the algorithm.

Sampling strategy. In LUCIE, identification of the elite is performed by sequentially sampling individuals among the population. To quickly identify elite individuals, we adopt the same strategy as [24]. They provide arguments suggesting that sampling both the individual of lowest lower bound in $\text{High}^{g,t}$ and the individual of highest upper bound in $\text{Low}^{g,t}$ is efficient. We define them as

$$\begin{aligned} h^t &\stackrel{\text{def}}{=} \operatorname*{arg\,min}_{i \in \text{High}^{g,t}} \hat{f}_i^{g,t} - \beta\left(u_i^{g,t}, t\right), \\ l^t &\stackrel{\text{def}}{=} \operatorname*{arg\,max}_{i \in \text{Low}^{g,t}} \hat{f}_i^{g,t} + \beta\left(u_i^{g,t}, t\right). \end{aligned} \tag{1}$$

These two individuals are the most ambiguous regarding elitism. Overall, when a new generation starts, LUCIE samples at least once each new individual and then samples sequentially pairs of individuals according to Equation (1) until termination.

Termination criterion. The algorithm terminates once the population is *separated* between the estimated elite and the rest with high certainty. This corresponds to the case where the lower-bounds of the individuals in $\text{High}^{g,t}$ are all above the upper-bounds of the individuals in $\text{Low}^{g,t}$. An error margin of ϵ is allowed. Formally, the algorithm stops if the following inequality is verified:

$$\hat{f}_{l^t}^{g,t} + \beta\left(u_{l^t}^{g,t}, t\right) < \hat{f}_{h^t}^{g,t} - \beta\left(u_{h^t}^{g,t}, t\right) + \epsilon. \tag{2}$$

Once this criterion is reached, the EA carries the estimated elite $\text{High}^{g,t}$ over to the next generation $g + 1$, where λ new candidates are created to populate Ind^{g+1}, and the process is repeated.

4 THEORETICAL ANALYSIS

This section discusses LUCIE's theoretical properties. Theorem 4.1 states the conditions on β under which LUCIE is correct with high probability, that is, upon termination, it identifies a set of (ϵ, μ)-optimal arms with high probability. In turn, these conditions allow deriving the analytical form of the confidence interval β in Equation (4). In Theorem 4.2, we show that, using this bound β, LUCIE converges to a solution in a finite number of fitness samples that is log-linear in the problem complexity. Then, in Theorems 4.3 and 4.4, we show that LUCIE converges in a finite number of generations on the stochastic version of the OneMax and LeadingOnes problems. Importantly, we show that this is achieved regardless of the degree of stochasticity, meaning that convergence is ensured, even with highly stochastic fitness functions.

4.1 Correctness and Sample Complexity

The first result establishes a condition on β for LUCIE to return (ϵ, μ)-optimal solutions. We define $[a \wedge b] \stackrel{\text{def}}{=} \max\{a, b\}, \forall a, b \in \mathbb{R}$. The results presented in this section are extensions of the LUCB [24] properties to the evolutionary setting.

Theorem 4.1 (Proof of correctness). *Let g be the current generation and $\beta : \mathbb{N}^2 \to \mathbb{R}^+$ a function such that*

$$\sum_{i=1}^{n}\sum_{t=1}^{\infty}\sum_{u=1}^{\infty}\exp\left(-2u\beta\left(u,t\right)^2\right) \leq \delta, \quad (3)$$

if LUCIE terminates (Equation 2), the probability of returning a non-(ϵ,μ)-optimal individual is at most δ, with $0 < \delta \leq 0.5$.

All proofs are deferred to the Appendix. Choosing β such that Equation (3) is verified allows bounding by δ the probability of failing (*i.e.*, returning a non-(ϵ,μ)-optimal solution). Equation (3) leads to the following choice of β:

$$\beta\left(u,t\right) \overset{\text{def}}{=} \sqrt{\frac{1}{2u}\ln\left(\frac{nkt^4u^2}{\delta}\right)}, \quad (4)$$

where $k = \pi^6/540 \approx 1.8$. This resulting bound is different from the LUCB bound in two matters. First, the appearance of u^2 in the log, which comes from the possibility for the number of samples u to be arbitrarily large. Secondly, and most importantly, this number of samples is the number of times an individual has been sampled *since its creation*. Therefore, the value of u may be large and exceed the number of steps t in the case of an elite individual persisting between generations. This is an advantage of LUCIE: past samples are retained across generations, keeping the uncertainty of fitness estimates low. Using the bound of Equation (4), we obtain that the algorithm reaches the termination criterion of Equation (2) in a finite number of steps, polynomial in the problem complexity. Following [24], the problem complexity $H^{g,\tau}$ is defined as the proximity between elite and non-elite expected fitnesses. We shall first introduce the notion of *domination gap* Δ_i before defining $H^{g,\tau}$:

$$\Delta_i \overset{\text{def}}{=} \begin{cases} f_i - \max\limits_{j\in \text{Bot}^g} f_j & \text{if } i \in \text{Top}^g, \\ \min\limits_{j\in \text{Top}^g} f_j - f_i & \text{if } i \in \text{Bot}^g. \end{cases}$$

Intuitively, the smaller the domination gaps, the less distinction between elite and non-elite members, *i.e.*, the harder the problem is. This yields the following definition of the problem complexity:

$$H^{g,\tau} \overset{\text{def}}{=} \sum_{i\in \text{Ind}^g} \frac{1}{\left[\tau \wedge \Delta_i^2\right]^2}.$$

$H^{g,\tau}$ aggregates all the domination gaps of the individuals, capped by a parameter value τ. This implies that $H^{g,\tau}$ is bounded between n and n/τ^2. τ is a tolerance threshold beyond which individuals are considered to yield equivalent expected fitness. We use $\tau = \epsilon/2$ in the sample complexity result below.

Theorem 4.2 (Expected Sample Complexity). *At generation g, the expected sample complexity of LUCIE is $O\left(\left(H^{g,\frac{\epsilon}{2}}\ln\left(\frac{H^{g,\frac{\epsilon}{2}}}{\delta}\right)\right)^{\frac{1}{\gamma}}\right)$, with γ any constant value such that $0 < \gamma < 0.57$.*

This result guarantees that the algorithm will converge in a finite number of steps. Finally, as β verifies Equation (3), LUCIE is able to select (ϵ,μ)-optimal individuals in an expected number of steps which is polynomial in the problem complexity $H^{g,\frac{\epsilon}{2}}$.

4.2 Stochastic OneMax and LeadingOnes

We now characterize sample complexity in the noisy OneMax and LeadingOnes optimization problems. This study has been carried out for the case of (1+1) EA by [16] and we here realize its counter part for the case of LUCIE.

OneMax. The OneMax(x) function counts the number of 1s in a vector of bit x of length N. Instead of the true fitness f_i, sampling an individual's fitness yields a noisy observation $O\left(i\right)$, which is a random variable following a so-called noise model. [16] make two assumptions regarding the noise model. First, they assume the larger the true difference in fitness, the more likely the observations are correctly ordered. Formally, this is written H1: $\forall j \leq l < N, \forall i_1, i_2 \in \text{Ind}, \mathbb{P}(O\left(i_1\right) < O\left(i_2\right) | f_{i_1} = j, f_{i_2} = l+1) \leq \mathbb{P}\left(O\left(i_1\right) < O\left(i_2\right) \mid f_{i_1} = l, f_{i_2} = l+1\right)$.

The second assumption is more restrictive. The probability of correctly ordering consecutive fitness individuals must not be *too low*:

H2: $\forall l < N, \forall c \in (0, \frac{1}{9}), \forall i_1, i_2 \in \text{Ind}$,

$$\mathbb{P}\left(O\left(i_1\right) < O\left(i_2\right) \mid f_{i_1} = l, f_{i_2} = l+1\right) \geq 1 - c\frac{N-l}{N}.$$

H2 has a direct impact on the bearable level of noise by (1+1) EA, which is expected. Indeed, in (1+1) EA, elite selection is performed on a single fitness observation, which is less likely to be correct if H2 is not verified. This restrictive assumption makes (1+1) EA unfit for stochastic OneMax with high levels of noise. For instance, if the noise on f_i follows a centered Gaussian law, H2 requires the variance σ^2 to be $O\left(\log(N)/N\right)$ to ensure convergence in a polynomial number of generations. In the following result, we show that LUCIE converges in the same number of generations as (1+1) EA while not requiring condition H2. Instead, we set a condition on the values of ϵ and δ.

Theorem 4.3. *For any noise model verifying H1, setting $\epsilon < 1$ and $\delta \leq kc/N$, (1+1) LUCIE optimizes the stochastic OneMax problem in $O\left(N\ln\left(N\right)\right)$ number of generations.*

LeadingOnes. A similar result can be achieved for the LeadingOnes function which counts consecutive 1s from the first bit in a binary vector of length N. Again, an observation $O\left(i\right)$ is returned instead of the true fitness of individual i. We also introduce the bits vectors x_l^{opt} and x_l^{pes} for $l \leq N$, respectively a vector of 1s except one single 0 at position $l+1$, and a vector of 1s except for $N-l$ 0s at the end. [16] make three assumptions (H3, H4, and H5) regarding the noise model. Condition H3 states that for an individual of true fitness l, the observation is drawn according to a distribution between $O(x_l^{\text{opt}})$ and $O(x_l^{\text{pes}})$ with respect to stochastic dominance. Condition H4 imposes that the larger the true difference in fitness, the more likely the observations are correctly ordered. Formally, H4: $\forall j \leq l < N, \forall i_1, i_2 \in \text{Ind}, \forall s \in \{\text{opt}, \text{pes}\}, \mathbb{P}(O(x_j^s) < O(x_{l+1}^s)) \leq \mathbb{P}(O(x_k^s) < O(x_{l+1}^s))$. Finally, assumption H5 is more restrictive. Similarly to the case of OneMax, the probability of correctly ordering consecutive fitness individuals must not be *too low*:

H5: $\forall l \leq N, \forall c \in]0, \frac{1}{12}[, \forall i_1, i_2 \in \text{Ind}$,

$$\mathbb{P}\left(O\left(i_1\right) < O\left(i_2\right) \mid i_1 = x_l^{\text{opt}}, i_2 = x_{l+1}^{\text{pes}}\right) \geq 1 - \frac{c}{lN},$$

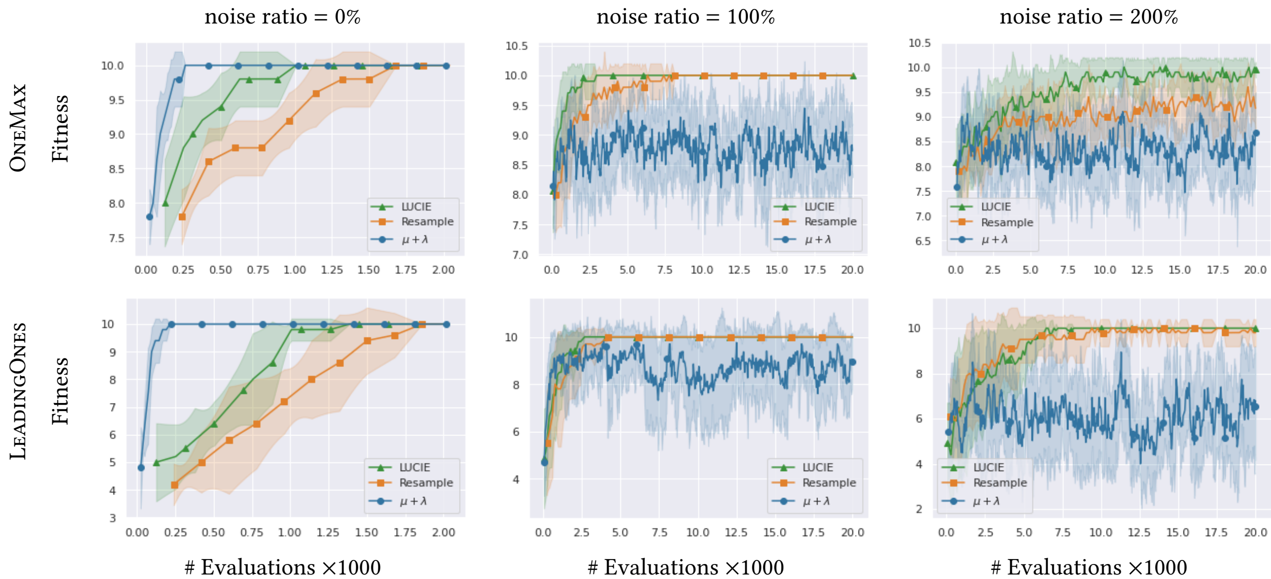

Table 1: Results for OneMax and LeadingOnes under posterior uniform noise.

H5 has the same impact on the noise model as H2. Again, as observations are single samples in the case of (1+1) EA, the probability for those samples to yield a wrong ordering must not be too high for the algorithm to converge in a finite number of generations. We translate this restriction H5 in terms of ϵ and δ parameters for LUCIE in the following result.

Theorem 4.4. *For any noise model verifying H3 and H4, setting* $\epsilon < 1$ *and* $\delta \leq k/12N^2$*, (1+1) LUCIE optimizes the stochastic LeadingOnes problem in* $O\left(N^2\right)$ *number of generations.*

Overall, to provably solve the noisy OneMax and LeadingOnes problems in a finite number of generations, (1+1) EA depends on two restrictive assumptions H2 and H5 on the level of noise [16]. LUCIE can overcome those assumptions by selecting (ϵ, μ)-optimal individuals with high probability (Theorem 4.1). This translates into a specific choice of ϵ and δ parameters.

5 EXPERIMENTAL ANALYSIS

In this section, we empirically validate the advantage of LUCIE in terms of efficient and accurate elite selection[1]. First, we validate our theoretical results on OneMax and LeadingOnes with experiments that apply uniform and Gaussian posterior noise to fitness. Secondly, we explore the use of LUCIE on more complex objective functions by studying neuroevolution for control policy optimization. We evolve neural networks which control actions in three different classic robotic tasks. This increases the complexity in the objective function and uses a much larger genotype than the OneMax and LeadingOnes problems. Due to the problem complexity and genome size, we do not compare with surrogate modelling technique but rather with resampling methods.

In all experiments, we benchmark three algorithms, namely the $(\mu + \lambda)$ EA, a resampling EA (which also has a $(\mu + \lambda)$ population size), and LUCIE. All algorithms use tournament selection for selecting offspring's parent individuals with a tournament size of 3. We write n_{RS} the number of samples per-individual used for selection in resampling EA. Hence, $(\mu{+}\lambda)$ EA performs $\mu{+}\lambda$ evaluations per generations while resampling EA performs $n_{RS}{\times}\lambda$ evaluations. In LUCIE, convergence to the exact (ϵ, μ)-optimal individuals is not always desirable, typically, in the case where individuals all have similar expected fitness. To prevent the algorithm from getting stuck in such a case, we max out the number of evaluation per generation with a parameter B. The termination criterion is thus reached either with Equation (2) or by reaching B evaluations. Confidence bounds were scaled using a multiplying factor α to account for the variation in fitness value ranges between different environments, as Equation (4) and (2) assume that fitness values are in $[0, 1]$. In the binary optimization experiment, we kept α as a constant parameter. In neuroevolution, α was defined as the difference of the largest fitness value from the previous generation, $f_{\max} = \max_i(\hat{f}_i^{g-1})$, and the smallest, $f_{\min} = \min_i(\hat{f}_i^{g-1})$, yielding $\alpha \overset{\text{def}}{=} (f_{\max} - f_{\min})$. In practice, this corresponds to the realistic case where maximum fitness of the problem is unknown. In this case, we also scale the value of ϵ so that all members of the stopping criterion inequality (Equation (2)) have the same scale.

	μ	λ	B	ϵ	δ	α	n_{RS}
(a)	6	18	120	1	0.1	$f_{\max} - f_{\min}$	10
(b)	4	16	10^4	$f_{\max} - f_{\min}$	0.1	$f_{\max} - f_{\min}$	10

Table 2: Parameters for (a) binary optimization and (b) neuroevolution experiments.

[1] Code available at github.com/TemplierPaul/pyUCEA

5.1 Binary optimization

We optimize the stochastic OneMax and LeadingOnes functions under several posterior noise models. Individuals are represented by a bits vector of size $N = 10$, leading to a maximum possible expected fitness of 10. Creation of offsprings is performed via a mutation operator, flipping each bit with probability $1/10$. Gaussian noise is drawn from a normal distribution of standard deviation σ. Uniform noise is drawn uniformly in $[-10, 10]$ and then multiplied by a noise ratio that is a parameter. In both cases, a noise sample is added to the true individual's fitness each time a fitness is sampled. We report results for noise parameters of $\sigma \in \{0, 10, 30\}$ and a noise ratio in $\{0\%, 100\%, 200\%\}$. Table 2 reports the parameters for LUCIE, $(\mu + \lambda)$ EA and resampling EA. Shared parameters are common to all algorithms. Results are reported in Table 1 for uniform noise and in Appendix 6 for Gaussian noise. We show one standard deviation as confidence intervals obtained with 100 runs for both experiments.

In low noise regimes, we first observe that $(\mu+\lambda)$ EA shows the highest convergence speed while resampling EA shows the lowest. This is too be expected as computation is wasted on resampling for the latter. Lower Upper Confidence Intervals Elitism (LUCIE) mitigates this effect by dynamically triggering the end of a generation thanks to the stopping rule. Thus, it shows an in-between performance. Further tuning of the ϵ parameter would guarantee that LUCIE uses fewer evaluations per generation as the stopping condition would be reached faster. In medium noise regime, LUCIE outperforms the EAs baselines consistently. The resampling EA outperforms $(\mu + \lambda)$ EA as elite selection becomes difficult. The latter shows no-learning progress as the level of noise is too high. These effects are augmented in the high noise setting. As LUCIE is able to successfully adapt to all noise regimes, this makes it a good, versatile, candidate in face of uncertainty with that respect, while having to chose between resampling or not may hinder an EAs performance. Generally in those experiments, LUCIE featured less variance than any other EA. We believe that the stronger assessment of elite individuals it provides yields stability in learning. All those conclusions can be observed in both Gaussian and uniform noise, suggesting that LUCIE is robust to the type of noise it faces.

5.2 Neuroevolution

We evolve policies for three robotic control tasks, specifically CartPole balancing, the Acrobot swing-up problem [29], and an inverted Pendulum cart balancing. We use the OpenAI gym implementation of these problems [4]. Individuals are represented by a neural network parameterized by two dense layers of size 32 using tanh activation function with a linear output layer corresponding to the number of actions. Genes are continuous values initialized using a Glorot initialization [17]; mutation consists of the application of Gaussian noise with a standard deviation of 0.1, similar to the Genetic Algoritm neuroevolution in [6].

We report results for noise parameters of noise ratio in { 0%, 200%, 400%, 600%, 800% }. The algorithms parameters are summarized in Table 2 and shared parameters are common between tasks. Validation results are reported in Tables 3, 4 and 5 which show boxplots displaying quartiles as confidence intervals obtained with 30 runs for each experiment. Additional results are reported in Appendix 7. In low noise regimes, LUCIE features a similar final score as resampling EA, while being outperformed by $(\mu + \lambda)$ EA, except in the CartPole case where they all reach top expected fitness. In medium noise regimes, LUCIE reaches better final score as the EAs consistently. Like in binary evolution, resampling EA outperforms $(\mu + \lambda)$ EA as elite selection becomes difficult under higher levels of noise. Critically, LUCIE is able to converge to optimal solutions in extreme level of noise, reaching high scores up to 800% of noise ratio. Generally in most experiments, LUCIE featured less variance than classical EA. This observation reinforces our belief that accurate elite selection leads to stable learning, confirming results on OneMax and LeadingOnes.

6 CONCLUSION

We introduced LUCIE, an individuals selection method fit for elitism in EAs optimizing stochastic fitness functions. LUCIE casts the problem of selection to a bandit problem of best μ arms identification. An empirical mean fitness along with a confidence interval is maintained for each individual, permitting to stop the algorithm when (ϵ, μ)-optimal individuals are selected with probability at least $1-\delta$. This later property of LUCIE was demonstrated in our theoretical analysis. Furthermore, we formally showed that convergence in the stochastic versions of the OneMax and LeadingOnes problems was enabled for mild assumptions on the noise level. In comparison, standard EAs need strong requirements, linked to the noise variance, for convergence to be provably guaranteed.

To the best of our knowledge, the method of LUCIE is a novel application of bandit algorithm to individuals selection in evolution. Further, one could analyse application of other best μ arm identifications techniques to this setting [5], such as parent selection in addition to elitism. This could be an occasion to explore other aspects such that parallel evaluations of individuals.

Experimentally, we confirmed those results on OneMax and LeadingOnes under high posterior Gaussian and uniform noise. The algorithm outperforms both $(\mu + \lambda)$ EA and resampling EA in terms of learning speed, while maintaining a competitive performance in low noise regime. This latter fact suggests that LUCIE is versatile and could be used confidently without prior knowledge on the noise level. The algorithm also features minimal variance, yielding consistent results between runs. We also conducted neuroevolution experiments in the cartpole, acrobot, and inverted pendulum domain under posterior uniform noise. Noise level here was set to an extreme magnitude of up to 800% of the maximum reachable fitness. Here as well, LUCIE demonstrated faster convergence to the optimal solution and a minimal variance in the results compared to the EA baseline with and without resampling. We believe that LUCIE could be used for larger neuroevolution experiments, in genetic algorithms of larger sizes or evolutionary strategies. As LUCIE proposes a way to guarantee elite selection under noisy fitness evaluations, it has potential for use in various EAs in many contexts such as policy search, robotics, and multi-objective optimization.

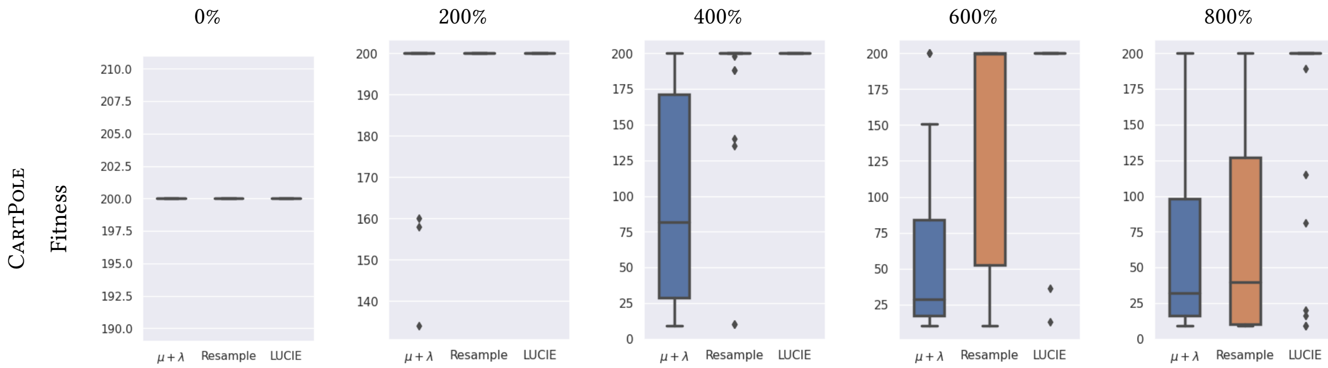

Table 3: Final validation fitness for CartPole neuroevolution under posterior uniform noise.

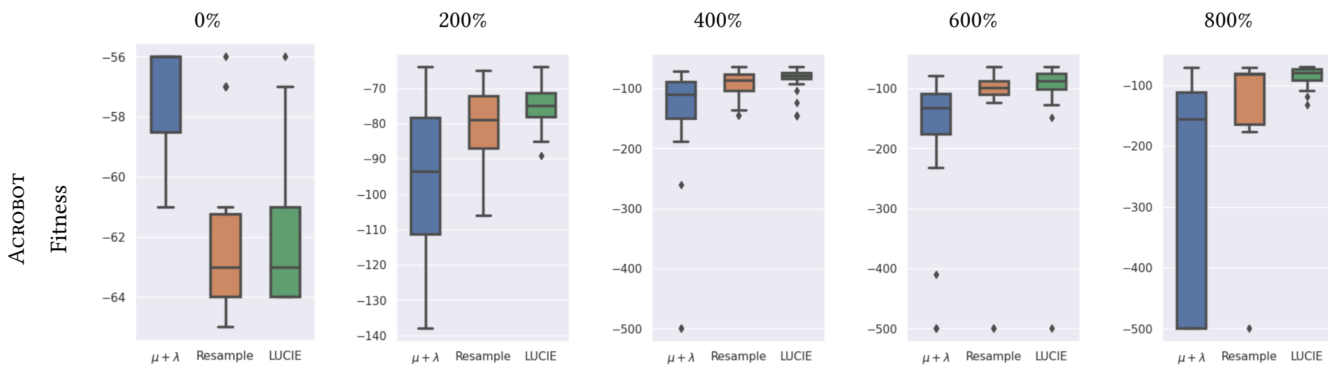

Table 4: Final validation fitness for Acrobot neuroevolution under posterior uniform noise.

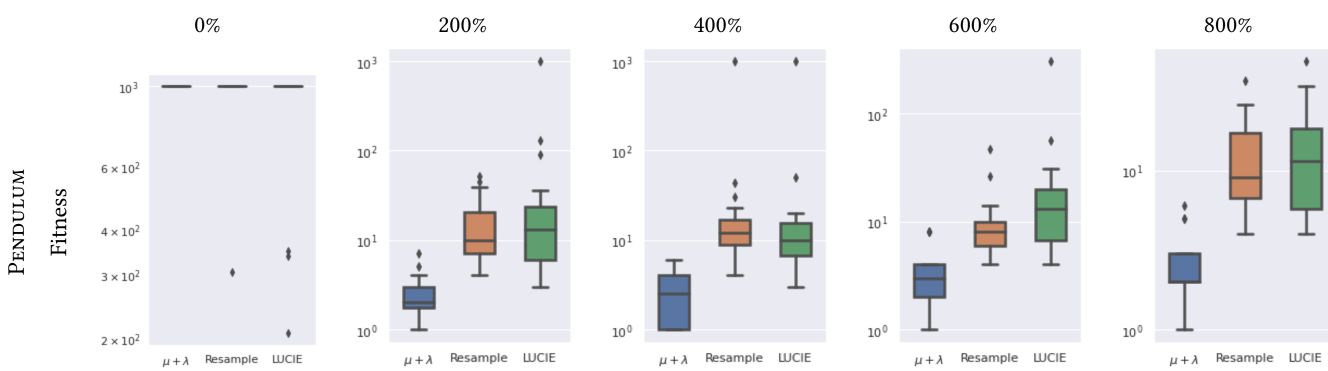

Table 5: Final validation fitness for Pendulum neuroevolution under posterior uniform noise.

ACKNOWLEDGMENTS

This work was funded by the Réseau Thématique de Recherche Avancée "Sciences et Technologies pour l'Aéronautique et l'Espace" de Toulouse (RTRA STAE) and was granted access to the HPC resources of CALMIP supercomputing center under the allocation 2022-P21001. We would like to thank the following researchers: Émilie Kaufmann for insightful discussions that brought inspiration and helped about the technicalities of the paper; Shivaram Kalyanakrishnan for helping with the examination of the LUCB algorithm; Sébastien Gerchinovitz for insightful discussions related to bandits. We also would like to thank François Lamothe, Valentin Guillet and the SuReLI (Supaero Reinforcement Learning Initiative) team for their support and fruitful discussions.

REFERENCES

[1] Youhei Akimoto, Sandra Astete-Morales, and Olivier Teytaud. 2015. Analysis of runtime of optimization algorithms for noisy functions over discrete codomains. *Theoretical Computer Science* 605 (2015), 42–50.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine learning* 47, 2 (2002), 235–256.

[3] Shumeet Baluja and Rich Caruana. 1995. Removing the Genetics from the Standard Genetic Algorithm. In *Proceedings of the 12th International Conference on Machine Learning (ICML 1995)*. Elsevier, 38–46.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[5] Sébastian Bubeck, Tengyao Wang, and Nitin Viswanathan. 2013. Multiple Identifications in Multi-Armed Bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*. PMLR, 258–265.

[6] Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. 2020. Scaling map-elites to deep neuroevolution. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 67–75.

[7] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.

[8] Kenneth A De Jong. 2006. *Evolutionary Computation: A Unified Approach*. MIT Press.

[9] Benjamin Doerr and Andrew M. Sutton. 2019. When resampling to cope with noise, use median, not mean. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2019)*. 242–248.

[10] Stefan Droste. 2004. Analysis of the (1+ 1) EA for a Noisy OneMax. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*. Springer, 1088–1099.

[11] Tobias Friedrich, Timo Kötzing, Martin S. Krejca, and Andrew M. Sutton. 2016. Graceful Scaling on Uniform versus Steep-Tailed Noise. In *Proceedings of the 14th International Conference on Parallel Problem Solving from Nature (PPSN 2016)*. Springer, 761–770.

[12] Tobias Friedrich, Timo Kötzing, Martin S. Krejca, and Andrew M. Sutton. 2016. The Compact Genetic Algorithm is Efficient Under Extreme Gaussian Noise. *IEEE Transactions on Evolutionary Computation* 21, 3 (2016), 477–490.

[13] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2018. Data-Efficient Design Exploration through Surrogate-Assisted Illumination. *Evolutionary Computation* 26, 3 (2018), 381–410.

[14] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2018. Data-efficient neuroevolution with kernel-based surrogate models. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2018)*. 85–92.

[15] Aurélien Garivier and Emilie Kaufmann. 2016. Optimal Best Arm Identification with Fixed Confidence. In *Proceedings of the 29th Conference on Learning Theory (COLT 2016)*. PMLR, 998–1027.

[16] Christian Gießen and Timo Kötzing. 2016. Robustness of Populations in Stochastic Environments. *Algorithmica* 75, 3 (2016), 462–489.

[17] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[18] Alexander Hagg. 2017. Hierarchical Surrogate Modeling for Illumination Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO Companion 2017)*. 1407–1410.

[19] JH Holland. 1975. Adaptation in Natural and Artificial Systems. *Ann Arbor: University of Michigan Press* (1975).

[20] Nick Jakobi, Phil Husbands, and Inman Harvey. 1995. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*. Springer, 704–720.

[21] Yaochu Jin. 2005. A comprehensive survey of fitness approximation in evolutionary computation. *Soft computing* 9, 1 (2005), 3–12.

[22] Yaochu Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 2 (2011), 61–70.

[23] Shivaram Kalyanakrishnan and Peter Stone. 2010. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*.

[24] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. 2012. PAC Subset Selection in Stochastic Multi-armed Bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, Vol. 12. 655–662.

[25] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit Algorithms*. Cambridge University Press.

[26] Frank Neumann, Mojgan Pourhassan, and Vahid Roostapour. 2020. Analysis of Evolutionary Algorithms in Dynamic and Stochastic Environments. In *Theory of Evolutionary Computation*. Springer, 323–357.

[27] Christiaan J Pretorius, Mathys C du Plessis, and John W Gonsalves. 2017. The transferability of evolved hexapod locomotion controllers from simulation to real hardware. In *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 567–574.

[28] Carl Edward Rasmussen and Christopher K. Williams. 2006. *Gaussian Processes for Machine Learning*. Vol. 2. MIT press Cambridge, MA.

[29] Mark W. Spong. 1995. The swing up control problem for the acrobot. *IEEE control systems magazine* 15, 1 (1995), 49–55.

[30] Dirk Thierens and David Goldberg. 1994. Convergence models of genetic algorithm selection schemes. In *International conference on parallel problem solving from nature*. Springer, 119–129.

[31] Holger Ulmer, Felix Streichert, and Andreas Zell. 2003. Evolution strategies assisted by Gaussian processes with improved preselection criterion. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, Vol. 1. IEEE, 692–699.

[32] Leslie G. Valiant. 1984. A Theory of the Learnable. *Commun. ACM* 27, 11 (1984), 1134–1142.