Journal of Artificial Intelligence Research 70 (2021) 891-921 Submitted 04/2020; published 03/2021

# On the Evolvability of Monotone Conjunctions with an Evolutionary Mutation Mechanism

**Dimitrios I. Diochnos** DIOCHNOS@OU.EDU
*University of Oklahoma*

## Abstract

A Bernoulli$(p)^n$ distribution $\mathcal{B}_{n,p}$ over $\{0,1\}^n$ is a product distribution where each variable is satisfied with the same constant probability $p$. Diochnos (2016) showed that Valiant's swapping algorithm for monotone conjunctions converges efficiently under $\mathcal{B}_{n,p}$ distributions over $\{0,1\}^n$ for any $0 < p < 1$. We continue the study of monotone conjunctions in Valiant's framework of evolvability. In particular, we prove that given a $\mathcal{B}_{n,p}$ distribution characterized by some $p \in (0, 1/3] \cup \{1/2\}$, then an evolutionary mechanism that relies on the basic mutation mechanism of a (1+1) evolutionary algorithm converges efficiently, with high probability, to an $\varepsilon$-optimal hypothesis. Furthermore, for $0 < \alpha \leq 3/13$, a slight modification of the algorithm, *with a uniform setup this time*, evolves with high probability an $\varepsilon$-optimal hypothesis, for every $\mathcal{B}_{n,p}$ distribution such that $p \in [\alpha, 1/3 - 4\alpha/9] \cup \{1/3\} \cup \{1/2\}$.

## 1. Introduction

Valiant (2009) introduced a framework for a quantitative approach to evolution, called *evolvability*. The idea is, roughly, that there is an *ideal behavior* in every environment and the feedback that the various organisms receive during evolution indicates how close their behavior is to ideal. Ultimately, evolvability aims at modeling and explaining mechanisms that allow near-optimal behavior of organisms while exploiting realistic computational resources. Due to a result by Feldman (2008), evolvability is equivalent to learning in the *correlational statistical query* (CSQ) model (Bshouty & Feldman, 2002). Thus, evolvability algorithms correspond to a special type of local search learning algorithms that fall under the umbrella of the *probably approximately correct* (PAC) model of learning (Valiant, 1984). In fact Valiant (2013) gives a broad exposition of such algorithms for evolution, which he calls *ecorithms*, and discusses them within the context of computational complexity and computational learning theory. Watson and Szathmáry (2016) have an interesting related discussion on the connections between computational learning and evolution.

A key challenge for machine learning (and more broadly, for artificial intelligence) algorithms, is that of *brittleness*. That is, typically many artificial intelligence systems *fail* when tested outside of some narrow domain for which they have been designed; such discussions go back to expert systems (see, e.g., Duda & Shortliffe, 1983). John Holland (1986) argues on the use of *genetic algorithms* in order to handle brittleness. Valiant (2013) also mentions brittleness as a challenge that needs to be addressed by ecorithms. Both Holland and Valiant argue that learning is needed in order to tackle brittleness.

As a consequence, machine learning algorithms that can cope with noise are more desirable as they are more realistic for practical purposes. Quinlan (1986) conducted one of the first wide experimental studies on various types of noise. However noise has been studied

©2021 AI Access Foundation. All rights reserved.

extensively within PAC learning (see, e.g., Laird, 1988; Sloan, 1995) and in particular an important framework is the *statistical query* model (Kearns, 1998). It is within this framework that we find the CSQ model and due to Feldman's result, evolvability. Ecorithms, by the very nature of the model, have to deal with noisy estimates. Such estimates represent the *goodness of fit* of individuals within environments and are computed by a limited amount of interaction that individuals have with the environment.

**Importance of the Problem and Motivation.** Learning monotone conjunctions is arguably the most natural problem studied in the theory of machine learning as it encapsulates, in a very basic form, a class of functions that determine among a pool of features which ones should be included as relevant for a prediction mechanism. Conjunctions are the building blocks for Disjunctive Normal Form formulae (DNFs) which can represent *any* Boolean function, and monotone conjunctions are the case of conjunctions that contain no negations and therefore perhaps the simplest non-trivial class of Boolean functions. Hence it is natural to explore the learnability (evolvability) of such a class of functions in different (restricted) learning setups hoping that such mechanisms can extend to richer classes of functions and/or distributions. The current paper is contributing in this direction. Diochnos and Turán (2009, Example 1) have shown that the simple mechanism that adds a variable, removes a variable or swaps a variable for another one in the hypothesis, may get stuck in local optima under product distributions. This phenomenon motivates the work on global search operators that have the potential of taking us away from local optima and it is such an operator that we study in this paper.

**Brief Overview of Contributions.** We provide two ecorithms for a class of product distributions such that the first ecorithm adapts to each distribution in the class, whereas the second ecorithm uses a uniform setup for the entire class of distributions of interest. Both methods use a global search operator.

### 1.1 Models Related to Our Work

We first summarize the models that are connected to the work presented in this paper.

#### 1.1.1 The Probably Approximately Correct (PAC) Model of Learning

Valiant (1984) defined what is perhaps the default model in computational learning. In this model the learner typically knows the class of functions $\mathcal{C}_n$, called *concept class*, from where a *target function* $c$ is drawn. (The subscript $n$, for example in $\mathcal{C}_n$, indicates the dimension of function inputs; e.g., vectors in $\mathbb{R}^n$, or truth assignments over $\{0,1\}^n$.) The goal of the learner is to create a *hypothesis* $h$ that when asked about predicting the labels of various inputs (called *instances*), $h$ has error rate at most $\varepsilon$ w.r.t. $c$. Part of the design of the learning algorithm is the selection and *encoding* of the class of functions $\mathcal{H}$, called the *hypothesis class*, among which the learner will select a hypothesis $h$. It is important that the learning process takes place in time polynomial w.r.t. $1/\varepsilon$, $1/\delta$, and $n$. Ultimately the learner is allowed to find an $h$ that has error rate at most $\varepsilon$, not necessarily always, but with probability at least $1-\delta$. The parameters $\varepsilon$ and $\delta$ give the name to the model; the hypothesis generated is *approximately correct* but only *probably*. Thus, we have the *probably approximately correct (PAC)* model of learning.

In the standard variant of the model, instances are drawn independently and identically distributed (iid) according to some underlying distribution $D_n$ that is *unknown* to the learner, then those instances are labeled according to the target function $c$, and ultimately this sequence of pairs of instances and labels is fed into the learner as the set of training examples $S$. The learner uses this training set $S$ in order to form a hypothesis $h$ and the assumption is that the same probability distribution $D_n$ is going to govern the instances that will be drawn during testing and for which the hypothesis $h$ will predict their labels. The PAC requirement is that $h$ will behave well during testing (i.e., $h$ will have error rate at most $\varepsilon$) *regardless of the underlying distribution $D_n$ and despite the fact that $D_n$ is unknown to the learner*. Hence, this variant of PAC learning is called *distribution-independent*, or *distribution-free*.

In other variants, starting with the work of Benedek and Itai (1991), the learner may know the exact underlying distribution $D_n$ that governs the training and testing phases (e.g., uniform over $\{0,1\}^n$), or perhaps may know that the underlying distribution $D_n$ is some distribution from a broader class of distributions $\mathcal{D}$ but $D_n$ is not arbitrary (e.g., $\mathcal{D}$ is the class of product distributions over $\{0,1\}^n$). In these variants the learner may use the information about the underlying distribution and design an algorithm that is effective for the specific distribution, or class of distributions. When the algorithm is tailored to a specific distribution $D_n$, or when the algorithm is parameterized in such a way so that it can adapt to every $D_n \in \mathcal{D}$, then we talk about *distribution-specific learning*. In other cases, the algorithm designed might be effective for the entire class of distributions $\mathcal{D}$ without any parameterization. In this last case, the learner does not care about the exact underlying distribution $D_n$ (as long as $D_n \in \mathcal{D}$) and for this reason we have *distribution-free learning for a class of distributions*, a type of learning that falls somewhere in between distribution-specific and truly distribution-free.

### 1.1.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) use methods inspired by biology in order to find the point where an *unknown (fitness) function* attains its maximum value; thus, EAs fall under the broader umbrella of *black-box optimization methods*. The most well studied algorithm in this framework is the (1+1) EA shown in Algorithm 1; Droste, Jansen, and Wegener (2002) indicate several interesting results.

**Algorithm 1:** The (1+1) Evolutionary Algorithm

**Input:** A function $f$ to be optimized over $\{0,1\}^n$.
**Output:** A solution $x$, candidate for optimizing $f$.

```
1 x ← random string from {0,1}^n;
2 repeat
3 |   Compute x' from x by flipping each bit of x independently with probability 1/n.
4 |   Replace x by x' if f(x') ≥ f(x)
5 until some termination condition is met;
```

### 1.1.3 Evolving Programs

There is a natural extension of methods inspired by biology where the aim is to identify a function that maximizes some objective function rather than finding the input point where a function attains its maximum as is the case with EAs. This framework is called Genetic Programming (GP) and typically the programs that are being evolved have a tree-like structure (Koza, 1993). Since the work in this paper is about computational learning we are interested in approximating well some target function as is the case in GP. However, we will be able to use the mechanism shown in Algorithm 1 as we will see in Section 1.3.

### 1.1.4 Evolvability and Ecorithms

At a high level, ecorithms are local-search methods that achieve a PAC criterion. Perhaps the key difference between evolvability and traditional EAs/GP is the fact that noise is natural in evolvability as the functionalities that evolve over time realize their fitness through interaction with the environment (sampling); not by being able to interpret arbitrarily small differences of the fitness function. However, contrary to evolvability, with EAs and GP one usually wants to identify precisely the ideal behavior; not just an $\varepsilon$-approximation.

We will be evolving Boolean functions and this interaction of the evolved organisms (hypotheses) with the environment is a value that indicates how well the organism is approximating the *ideal behavior* (read, *target function* $c$ in standard learning terminology) for the environment. The value that represents the 'goodness of fit' of each hypothesis is obtained by drawing a random sample of rows of the truth table and then letting the organism $h$ know an aggregate value of how frequently $h$ agrees with the ideal behavior (target function) $c$ on these rows. Note that ecorithms take as input the underlying distribution $D_n$ and therefore the results obtained are typically about distribution-specific learning, or about distribution-free learning restricted to a class of distributions (as explained in the paragraph earlier about PAC learning). This is the case, e.g., in the work of Kanade, Valiant, and Vaughan (2010), in the work of Michael (2012), in the work of Angelino and Kanade (2014), and in the work of Diochnos (2016). Ecorithms also fit very well within the framework of *learning by distances* (Ben-David, Itai, & Kushilevitz, 1995); a framework that is equivalent to the CSQ model and was defined independently of the statistical query model of Kearns. Another closely related framework of learning to the model of evolvability is the model of learning with a restricted focus of attention (Ben-David & Dichterman, 1998). Below we provide more details for the model of evolvability and full definitions are available in Appendix A, or by Valiant (2009).

The truth values TRUE and FALSE are represented by 1 and $-1$ respectively. The objective function (or, *fitness* function) that guides the search is called *performance*. For a target $c$ and a distribution $D_n$ over $\{0,1\}^n$, the performance of a hypothesis $h$, called the *correlation* of $h$ and $c$, is,

$$\mathrm{Perf}_{D_n}(h,c) = \sum_{x\in\{0,1\}^n} h(x)\cdot c(x)\cdot \mathbf{Pr}_{x\sim D_n}(x) = 1 - 2\cdot \mathbf{Pr}_{x\sim D_n}(h(x)\neq c(x)) \,. \quad (1)$$

Note that an approximation error $\varepsilon$ for correlation implies misclassification error $\varepsilon/2$.

Evolution starts with some initial hypothesis and produces a sequence of hypotheses using a local-search procedure in the hypothesis space $\mathcal{H}$. Similarity between $h$ and $c$ in an

underlying distribution $D_n$ is measured by the *empirical performance* $\mathrm{Perf}_{D_n}(h, c, S)$ which is evaluated approximately by drawing a random sample $S$ (of size $|S|$) and computing

$$\mathrm{Perf}_{D_n}(h, c, S) = \frac{1}{|S|} \sum_{x \in S} h(x) \cdot c(x) \,. \tag{2}$$

The mutator function is responsible for generating the neighborhood $N(h)$ and selecting one hypothesis from $N(h)$ as the output for the next generation. For each hypothesis $h' \in N(h)$, the mutator first computes an empirical value of $\nu(h') = \mathrm{Perf}_{D_n}(h', c, S)$ and also associates each hypothesis $h'$ with a weight $\mathbf{Pr}_N(h, h')$. Then, based on a real constant $t$ called *tolerance* we obtain,

$$\begin{cases} \mathrm{Bene} &= \{h' \in N(h) \mid \nu(h') > \nu(h) + t\} \\ \mathrm{Neut} &= \{h' \in N(h) \mid \nu(h') \geq \nu(h) - t\} \setminus \mathrm{Bene} \end{cases} \tag{3}$$

The output of the mutator function is based on the rule:

- if $\mathrm{Bene} \neq \emptyset$ then output $h_1 \in \mathrm{Bene}$ with probability $\mathbf{Pr}_N(h, h_1) \,/\, \sum_{h' \in \mathrm{Bene}} \mathbf{Pr}_N(h, h')$,
- if $\mathrm{Bene} = \emptyset$ then output $h_1 \in \mathrm{Neut}$ with probability $\mathbf{Pr}_N(h, h_1) \,/\, \sum_{h' \in \mathrm{Neut}} \mathbf{Pr}_N(h, h')$.

Ultimately, the goal of the evolution is to produce, within a realistic time period (i.e., within $poly(1/\varepsilon, 1/\delta, n)$ generations), a hypothesis $h \in \mathcal{H}$ such that

$$\mathbf{Pr}\left(\mathrm{Perf}_{D_n}(h, c) < \mathrm{Perf}_{D_n}(c, c) - \varepsilon\right) < \delta \,. \tag{4}$$

### 1.2 Related Work

There is a plethora of previous work in the framework of evolvability (Valiant, 2009; Feldman, 2008, 2009, 2011, 2012; Diochnos & Turán, 2009; Kanade et al., 2010; Kanade, 2011; Michael, 2012; Angelino & Kanade, 2014; Valiant, 2014; Diochnos, 2016; Snir & Yohay, 2019a, 2019b). Regarding conjunctions, their evolvability follows by a result from Feldman (2008) for every fixed distribution within $\widetilde{\mathcal{O}}(n)$ generations; where $\widetilde{\mathcal{O}}(\cdot)$ ignores polylog factors. As this translation is not necessarily the most efficient or intuitive method in general, there is still interest in different evolution mechanisms. The evolvability of monotone conjunctions under the uniform distribution $\mathcal{U}_n$ with a swapping-type algorithm was initially shown by Valiant (2009). The analysis was simplified by Diochnos and Turán (2009) and the result was strengthened to general conjunctions[1] under $\mathcal{U}_n$ including target drift, by Kanade, Valiant and Vaughan (2010). Kanade (2011) introduced recombination, where it follows that conjunctions are evolvable in $\mathcal{O}\left((\log(n)/\varepsilon)^2\right)$ generations. Diochnos (2016) showed that monotone conjunctions are evolvable under Bernoulli$(p)^n$ distributions (product distributions where each variable has the same probability $p$ of being satisfied) for every $p \in (0, 1)$, in $\mathcal{O}(\log(1/\varepsilon))$ generations, by generalizing the swapping-type mechanism for $\mathcal{U}_n$. Snir and Yohay (2019a, 2019b) extended the evolvability model with horizontal gene transfer; in a theoretical work (Snir & Yohay, 2019b) the model with horizontal gene transfer was defined and ultimately the main result allowed the evolvability of conjunctions in

1. Evolving general conjunctions under $\mathcal{U}_n$ is attributed to B. Jacobson (see Kanade et al., 2010).

$O(1)$ generations for any fixed distribution, whereas in a more practical work (Snir & Yohay, 2019a) it was verified experimentally the acceleration in terms of the number of generations.

Conjunctions have also been studied within black box optimization by Ros (1992). However, the distribution-free results that were obtained by Ros ultimately rely on information such as the number of bits on which the hypothesis and the input differ. Such dependence on the input condition is considered unrealistic and is outside of the model of evolvability. Recently, Lissovoi and Oliveto (2019) studied monotone conjunctions within GP. Their work refers to the uniform distribution $\mathcal{U}_n$ over $\{0,1\}^n$, relies on initialization (namely, they start from the empty representation) and some of their results extend to the framework of evolvability; however, no concrete bounds on the tolerance and the sample size are mentioned. In addition, Kötzing, Neumann, and Spöhel (2011) examine a swapping-type mechanism for linear functions similar to Valiant's swapping algorithm.

Noise models have also been studied within EAs/GP. Droste (2004) discusses noise in the *prior noise model* while optimizing the OneMax function. In the prior noise model, the fitness oracle may, with some small probability, return the true fitness value but for an instance that is in the neighborhood of the queried instance. In the *posterior noise model* the true fitness value is corrupted by noise; e.g., by adding a small random value drawn from some fixed distribution. However, it is only in the last few years that noisy fitness values have been identified as a hot topic in the field (Friedrich & Neumann, 2017). There are several such recent results on either noise model (Astete-Morales, Cauwet, & Teytaud, 2015; Dang & Lehre, 2015; Prugel-Bennett, Rowe, & Shapiro, 2015; Gießen & Kötzing, 2016; Qian, Bian, Jiang, & Tang, 2019).

As a last remark, one can think of directions in optimization frameworks, where, while the problem spaces are different, nevertheless the primary motivation for the proposed solutions is relevant to the above lines of work. For example, the work of Mühlenbein and Mahnig (2001) proposed the UMDA algorithm which solves difficult multi-modal optimization problems. In addition, one can think of directions along the lines of *simulated annealing* (Cordón, de Moya Anegón, & Zarco, 2002; Cai & Shao, 2002; Gutjahr & Pflug, 1996).

### 1.3 Bringing the Different Models Together

We consider product distributions over $\{0,1\}^n$ such that *each variable follows the same Bernoulli($p$) distribution*. We call such distributions Bernoulli$(p)^n$ and denote them by $\mathcal{B}_{n,p}$. Hence, on a truth assignment of dimension $n$, a Bernoulli$(p)^n$ distribution $\mathcal{B}_{n,p}$ over $\{0,1\}^n$ is specified by the probability $p$ of setting each variable $x_i$ equal to 1. A truth assignment $(a_1,\ldots,a_n) \in \{0,1\}^n$ has probability $\prod_{i=1}^{n} p^{a_i} \cdot (1-p)^{1-a_i}$. We use $\mathcal{B}_n$ to denote a fixed Bernoulli$(p)^n$ distribution, omitting $p \in (0,1)$ for simplicity when it is clear from the context. The uniform distribution $\mathcal{U}_n$ is a Bernoulli$(p)^n$ distribution where $p = 1/2$. We note that Diochnos (2016) was using the term *binomial* distribution in order to refer to such a distribution $\mathcal{B}_{n,p}$ due to the fact that on a truth assignment drawn from such a distribution, the number of 1's are binomially distributed with parameters $p$ and $n$; however, the term Bernoulli$(p)^n$ is perhaps simpler and more fitting for such product distributions.

During the course of evolution we manipulate hypotheses that correspond to monotone conjunctions in every step. For some integer $q$, the distributions that we examine have the property that one can find optimal, or $\varepsilon$-optimal, approximations of any target function

(ideal behavior), when restricting the search to hypotheses that are composed of at most $q$ variables. We exploit this structural property and search only among such solutions. Hence, we call $q$ the *frontier* of our search. Throughout the paper $\widetilde{\mathcal{O}}(\cdot)$ will ignore poly-log factors in $n, 1/\varepsilon, 1/\delta$ and $q$, but not the frontier $q$ itself. ($q$ has logarithmic dependence on $1/\varepsilon$.)

Regarding the representation class $R_n$, we represent monotone conjunctions as sets of indices; the indices correspond to the variables that appear in the conjunctions, in some fixed ordering. Hence with $\Theta(\log n)$ bits we can represent each variable and thus a monotone conjunction that is composed of at most $q$ variables requires space $\mathcal{O}(q \log n)$. However, so that there is an easier connection to the results found in evolutionary programming literature, it is also convenient to think of the representation of a monotone conjunction $h$ as a binary string of length $n$, where a 1 in the $i$-th position indicates that the $i$-th variable appears in $h$; after all, going back and forth between these two representations can be done efficiently (polynomial time). In other words, for a binary string $\sigma = b_1 b_2 \dots b_n$, let $POS(\sigma) = \{i \mid b_i = 1\}$. Then, we can make a direct correspondence between bit strings and monotone conjunctions:

$$\sigma = b_1 b_2 \dots b_n \longmapsto \bigwedge_{i \in POS(\sigma)} x_i \,.$$

For two binary strings $\sigma$ and $\sigma'$ of length $n$, let $d_n(\sigma, \sigma') = \sum_{i=1}^{n} |\sigma_i - \sigma'_i|$ be their *Hamming distance*, where $\sigma_i$ and $\sigma'_i$ is their $i$-th bit respectively. Thinking of the representation of monotone conjunctions as bitstrings of length $n$, for any two monotone conjunctions $h_1$ and $h_2$ later on we will be able to compute $d_n(h_1, h_2)$ even if technically we will be passing sets of indices as parameters.

In this work we will use the representation mentioned above together with the mechanism for the (1+1) EA shown in Algorithm 1 and mutate from one representation to another. However, there is a catch. Monotone conjunctions that are composed of about linear in $n$ variables may have exponentially small differences when looking at the performance and in order for an evolutionary mechanism to understand such differences, an exponential amount of examples would be needed; this is unrealistic for efficient evolvability.

To the rescue, we use the above-mentioned property of the frontier $q$: there is a monotone conjunction $\varepsilon$-close to the optimum having at most $q$ variables. Our modified mutation mechanism will thus reject representations that have more than $q$ variables; see Section 2.1 for the algorithm. Such a design decision is a typical difference between machine learning and evolutionary algorithms; in machine learning we are not exploring the capabilities and limitations of 'pure' optimization algorithms such as the (1+1) EA shown in Algorithm 1, but rather we try to devise algorithms, perhaps less elegant, that nevertheless produce an $\varepsilon$-optimal solution overcoming potential obstacles that 'pure' and more intuitive optimization methods would otherwise have in some cases. Domingos (2015) has an interesting related discussion and historical remarks about the emergence of the schism in the kind of research performed by people working in machine learning on one hand and people working in evolutionary algorithms on the other.

### 1.4 Contributions

We extend the work that deals with the evolution of monotone conjunctions – using a global-search operator. The two lines of work that are closer to our work in this paper are: the work by Diochnos (2016) and the work by Lissovoi and Oliveto (2019).

Theorem 1 below gives an informal description of the positive result of our paper that is obtained when the evolutionary mechanism is adapted to the specific value of $p$ that characterizes the underlying distribution $\mathcal{B}_{n,p}$.

**Theorem 1** (Informal Version of Theorem 4)**.** *Let $\mathcal{B}_{n,p}$ be a Bernoulli$(p)^n$ distribution over $\{0,1\}^n$ characterized by some $p \in (0, 1/3] \cup \{1/2\}$. Then, for any starting representation, our evolutionary mechanism evolves an $\varepsilon$-optimal solution efficiently with high probability, using at most $q = \lceil \log_{1/p}(3/\varepsilon) \rceil$ variables in the solution.*

Our first contribution is that the representation that we use together with the basic evolutionary mechanism used in EAs provides an intuitive framework for evolving Boolean functions; a property that is desirable in general and in particular was also explicitly noted to be absent by Lissovoi and Oliveto (2019). In addition, the mutation operator from the (1+1) EA shown in Algorithm 1 is a global-search operator that has the potential of taking us away form local optima.

Our second contribution is that we provide a new characterization of the fitness landscape under the Bernoulli$(p)^n$ distributions $\mathcal{B}_{n,p}$ that we explore (where $p \in (0, 1/3] \cup \{1/2\}$) and it is this new characterization that allows us to prove convergence using a *fitness-level* technique; there are several other lines of work that use a similar technique (Wegener & Witt, 2005; Sudholt, 2010; Corus, Dang, Eremeev, & Lehre, 2018).

Third, our algorithm by default uses the underlying distribution as a hint and adapts to an appropriate representation and related parameters (e.g., sample size). To this end, most of our results refer to the parameter $p$ of the distribution as a real algebraic number[2] and leave open the exact bit complexity of required related calculations. A different point of view would be to assume that the algorithms are endowed with appropriate values that are the integers or some appropriate fractions near the true values implied by operations on arbitrary real numbers (referring to the distribution, or the inputs $\varepsilon$ and $\delta$), and then one happens to examine the mechanism on a setup where these values correspond to. In any case, we also provide *a positive result with a uniform setup over a class of distributions* as stated by the following theorem.

**Theorem 2** (Informal Version of Theorem 8)**.** *Let $0 < \alpha \leq 3/13$. Let $\mathcal{I} = [\alpha, 1/3 - 4\alpha/9] \cup \{1/3\} \cup \{1/2\}$. Consider a Bernoulli$(p)^n$ distribution $\mathcal{B}_{n,p}$ over $\{0,1\}^n$ that is characterized by such a $p \in \mathcal{I}$. Then, for any starting representation, our proposed evolutionary mechanism evolves efficiently an $\varepsilon$-optimal solution with high probability, using at most $q = \lceil \log_2(3/\varepsilon) \rceil$ variables in the solution.*

2. A real algebraic number $\rho$ can be encoded in *isolating interval representation* using a polynomial $f$ with rational coefficients and an interval $[\alpha, \beta]$ such that $\alpha, \beta \in \mathbb{Q}$ so that $\rho$ is the unique root of $f$ in the interval $[\alpha, \beta]$. Usually it is assumed that $f$ is square-free; that is, $f(\alpha)f(\beta) < 0$. Ultimately, the bit complexity results of the computational problems at hand, are given with respect to the input degree of the polynomials and the maximum bitsizes of the rationals that describe the various real algebraic numbers that are part of the input (e.g., see, Yap, 2000).

Note that the frontier $q$ in Theorem 2 is independent of the exact value of $p$ that governs the underlying distribution $\mathcal{B}_{n,p}$, thus contrasting Theorem 1 where there is a dependence.

Fourth, Feldman (2012) showed that conjunctions are evolvable in a distribution-independent way with a quadratic loss function, while in an earlier work Feldman (2011) showed that conjunctions are not CSQ learnable distribution-independently using Boolean loss; the loss function of the current paper. In this context, Theorem 8 has an added value as the result lies somewhere between distribution-specific and truly distribution-free learning.

Fifth, Kalai and Vempala (2006) showed that within simulated annealing it is important to allow descendents with performance worse than that of the parent. Paul Valiant (2014) asked whether similar phenomena can arise in evolvability. While our algorithm does not fully achieve this goal, nevertheless it is the first ecorithm that allows partial random walks, *starting from any initial candidate solution*, and does not necessarily follow strictly beneficial steps until a near-optimal solution is formed – for example, this was the case in the work by Diochnos (2016). Also, while the work of Lissovoi and Oliveto (2019) allows partial random walks, nevertheless their convergence result is obtained after initializing the hypothesis to the empty representation.

## 2. Preliminaries

Given a set of Boolean variables $x_1, \ldots, x_n$, we assume that there is an unknown target $c \in \mathcal{C}_n$, a monotone conjunction of some of these variables. Let $\mathcal{C}_n$ be the concept class of all possible monotone conjunctions. For a threshold $q$, let $\mathcal{C}_n^{\leq q}$ be the set of conjunctions from $\mathcal{C}_n$ that contain at most $q$ variables. Furthermore, let $\mathcal{C}_n^{>q} = \mathcal{C}_n \setminus \mathcal{C}_n^{\leq q}$. Our hypothesis space will be $\mathcal{H} = \mathcal{C}_n^{\leq q}$. With $|h|$ we denote the *size* (or *length*) of a monotone conjunction $h$; the number of variables that are contained $h$. Figure 1 provides a pictorial view of the distinction between the concept class and the hypothesis space that we use.

By (3) the neighborhood $N$ is split in 3 parts. There are *beneficial*, *neutral*, and *deleterious* mutations. Thus, we need an oracle for computing

$$\Delta = \mathrm{Perf}_{D_n}\left(h', c\right) - \mathrm{Perf}_{D_n}\left(h, c\right) ,$$

and hence for a tolerance $t$, determine the set where $h' \in N$ lies. In Appendix B we have a brief discussion about different models that one can form for evolution, primarily based on how this performance difference $\Delta$ is provided to the evolved organisms. Now let,

$$c = \bigwedge_{i=1}^{m} x_i \wedge \bigwedge_{k=1}^{u} y_k \qquad \text{and} \qquad h = \bigwedge_{i=1}^{m} x_i \wedge \bigwedge_{\ell=1}^{r} w_\ell \,. \tag{5}$$

The $x$'s are *mutual* variables, the $y$'s are called *undiscovered* or *missing*, and the $w$'s are called *redundant* or *wrong*. Variables in the target $c$ are called *good*, otherwise *bad*. Given a size $q$ and an *extension* $\vartheta$, a hypothesis $h$ is *short* when $|h| \leq q$, *medium* when $q < |h| \leq q+\vartheta$ and *long* when $|h| > q + \vartheta$.

**Definition 1** (Best $q$-Approximation)**.** *A hypothesis $h$ is called a best $q$-approximation of $c$ if $|h| \leq q$ and $\forall h' \neq h, |h'| \leq q : Perf_{D_n}\left(h', c\right) \leq Perf_{D_n}\left(h, c\right)$.*

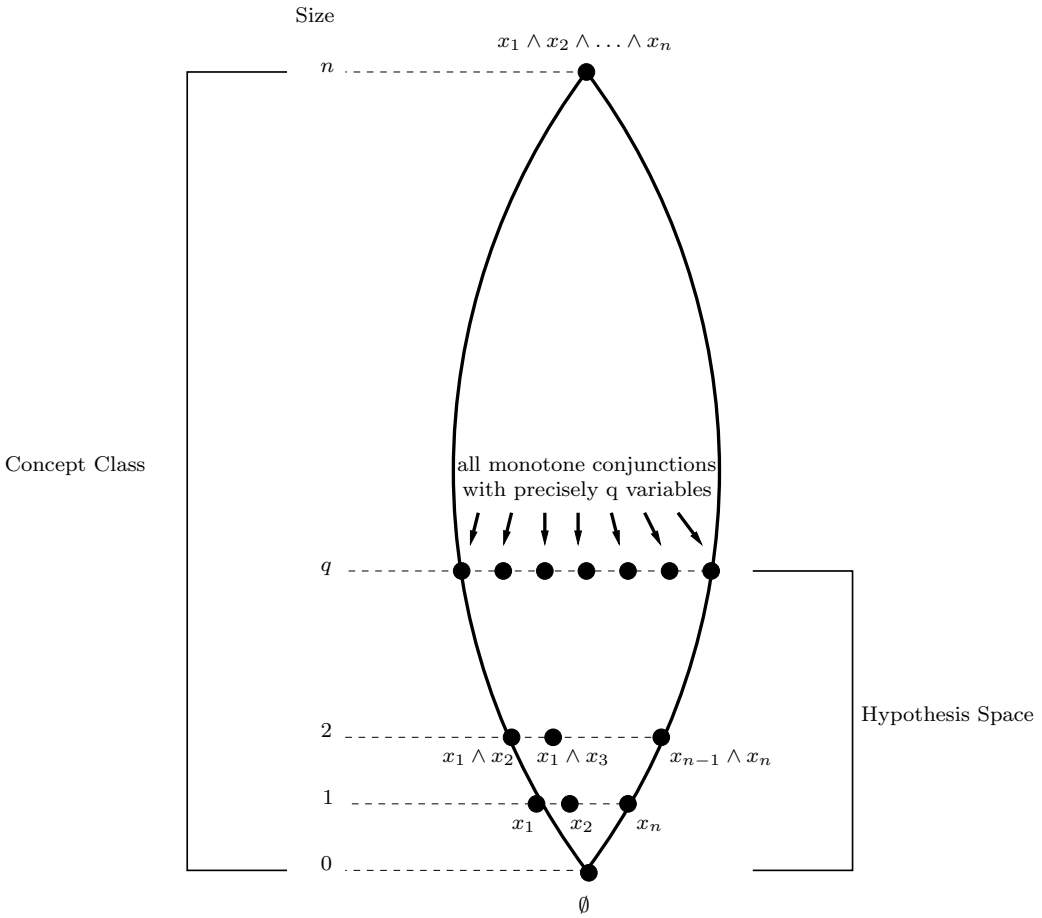



Figure 1: At the bottom of the figure we see the empty monotone conjunction that has size 0 corresponding to the identically true function, and at the very top we see the 'full' monotone conjunction of size $n$ where all the variables appear in the conjunction. Our hypothesis space is composed of monotone conjunctions that have size between 0 and $q$, for some threshold $q$ that we call the frontier of our search.

As mentioned in Section 1.3, we use sets of indices to represent monotone conjunctions and consider Bernoulli$(p)^n$ product distributions $\mathcal{B}_{n,p}$ over $\{0,1\}^n$. Now consider a target $c$ and a hypothesis $h$ as in (5). For a $\mathcal{B}_{n,p}$ distribution, (1) reduces to,

$$\mathrm{Perf}_{\mathcal{B}_n}(h,c) = 1 - 2p^{m+r} - 2p^{m+u} + 4p^{m+r+u}\,. \tag{6}$$

We will use $U = p^u$ for the weight of the subcube of the undiscovered variables. For a target $c$, we partition the hypothesis space in three parts; that is, we set $\mathcal{H} = \mathcal{H}_{<1/2} \cup \mathcal{H}_{1/2} \cup \mathcal{H}_{>1/2}$. $\mathcal{H}_{<1/2}$ refers to hypotheses for which $U < 1/2$, $\mathcal{H}_{1/2}$ refers to hypotheses for which $U = 1/2$ and $\mathcal{H}_{>1/2}$ refers to hypotheses for which $U > 1/2$. Figure 2 gives a pictorial example of this partitioning of $\mathcal{H}$ for a target monotone conjunction $c$ that is short under the uniform distribution over $\{0,1\}^n$. Ultimately, this partitioning gives rise to the different phases of the convergence of our methods. For example, for a short target function $c$ as

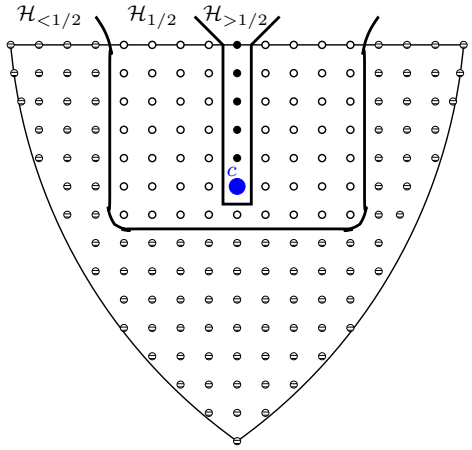



Figure 2: Partitioning of the hypothesis space $\mathcal{H}$ in $\mathcal{H}_{<1/2}$, $\mathcal{H}_{1/2}$, and $\mathcal{H}_{>1/2}$. Similarly to Figure 1, at the bottom of the figure we have the hypothesis that corresponds to the empty monotone conjunction and as we proceed towards the top we encounter monotone conjunctions that have progressively larger sizes, until we reach size $q$ at the very top, which is the frontier of our search. The large solid dot near the center of the hypothesis space is the target function $c$ that we want to learn. In this particular figure we assume that the underlying distribution is uniform over $\{0,1\}^n$ and thus the dots above $c$ correspond to hypotheses that belong to $\mathcal{H}_{>1/2}$ that have size larger than $|c|$ and moreover contain all the variables that appear in $c$.

in Figure 2, the evolved hypothesis will be proceed within $\mathcal{H}_{<1/2}$ with progressively larger sizes, then mutate from $\mathcal{H}_{<1/2}$ to $\mathcal{H}_{1/2}$, then mutate from $\mathcal{H}_{1/2}$ to $\mathcal{H}_{>1/2}$, and ultimately identify $c$ within $\mathcal{H}_{>1/2}$. (It is also possible that our hypothesis will mutate from $\mathcal{H}_{<1/2}$ to $\mathcal{H}_{>1/2}$ without ever visiting a hypothesis in $\mathcal{H}_{1/2}$.) Such separating groups of hypotheses are known as *fitness levels* and we can express this notion as follows. Under a distribution $D_n$, for a target $c$, and two sets of hypotheses $\Phi$ and $\Psi$, we write $\Phi \nrightarrow \Psi$ to indicate that

$$(\forall h_1 \in \Phi)(\forall h_2 \in \Psi)[\mathrm{Perf}_{D_n}(h_1, c) > \mathrm{Perf}_{D_n}(h_2, c)]\,.$$

Thus the hypotheses in $\Phi$ are in a 'higher' fitness level compared to the hypotheses found in the set $\Psi$.

With $\log_{1/p}(x)$ we denote the logarithm of $x$ in base $1/p$, where $0 < p < 1$. With $\mathbb{Q}$ and $\mathbb{R}_{alg}$ we denote respectively the fields of rational and real algebraic numbers. $H_j$ denotes the $j$-th harmonic number; that is, $H_j = \sum_{i=1}^{j} 1/i$.

Our proofs on the complexity analysis will also be using the following fact.

**Proposition 1** (Hoeffding's Bound; Hoeffding, 1963)**.** *Let $X_1, \ldots, X_R$ be $R$ independent random variables, each taking values in the range $\mathfrak{I} = [\alpha, \beta]$. Let $\mu$ denote the mean of their expectations. Then* $\mathbf{Pr}\left(\left|\frac{1}{R}\sum_{i=1}^{R} X_i - \mu\right| \geq \epsilon\right) \leq 2e^{-2R\epsilon^2/(\beta-\alpha)^2}$ .

### 2.1 The Algorithm

Algorithm 2 presents the mutator function. For a current hypothesis $h$, the evolutionary operator `Mutate` generates one candidate hypothesis $h'$ by first initializing $h'$ to $h$ and then

**Algorithm 2:** Mutator function of the (1+1) EA for Bernoulli$(p)^n$ product distributions.

**Input:** $n \in \mathbb{N}^*$, $p \in (0, 1/3] \cup \{1/2\}$, $\delta \in (0,1)$, $\varepsilon \in (0,2)$, $h \in \mathcal{H} = \mathcal{C}_n^{\leq q}$
**Output:** a new hypothesis

1 $q \leftarrow \lceil \log_{1/p}(3/\varepsilon) \rceil$; ; /* set the frontier */
2 $h' \leftarrow$ `Mutate`$(h)$; ; /* perform the mutation */
3 **if** $|h'| \leq q$ **then** $N \leftarrow \{h'\}$; /* set the neighborhood */
4 **else return** $h$;
5 **if** $0 < p < 1/3$ **then**
6 $\quad t \leftarrow p^{q-1} \cdot \min\{4p^q/3, 1-3p\}$ ; /* set the tolerance */
7 $\quad \delta_s \leftarrow \delta^2/(126en^2q)$ ; /* set the confidence for estimating the performance */
8 **else if** $p = 1/3$ **then**
9 $\quad t \leftarrow 2 \cdot 3^{-1-2q}$ ; /* set the tolerance */
10 $\quad \delta_s \leftarrow \delta^2/(126en^2q)$ ; /* set the confidence for estimating the performance */
11 **else**
12 $\quad t \leftarrow 2^{-2q}$; ; /* set the tolerance */
13 $\quad \delta_s \leftarrow \delta^2/(142en^2q)$ ; /* set the confidence for estimating the performance */
14 $\epsilon_s \leftarrow t/2$; ; /* set the approximation-error bound for estimating the performance */
15 $\nu_h \leftarrow$ `Perf`$(p, h, \epsilon_s, \delta_s)$ ; /* estimate empirically the performance */
16 $\nu_{h'} \leftarrow$ `Perf`$(p, h', \epsilon_s, \delta_s)$; ; /* estimate empirically the performance */
/* apply the selection mechanism below and return the appropriate hypothesis */
17 **if** $\nu_{h'} > \nu_h + t$ **then return** $h'$;
18 **else if** $\nu_{h'} \geq \nu_h - t$ **then return** `USelect`$(\{h\} \cup \{h'\})$;
19 **else return** $h$;

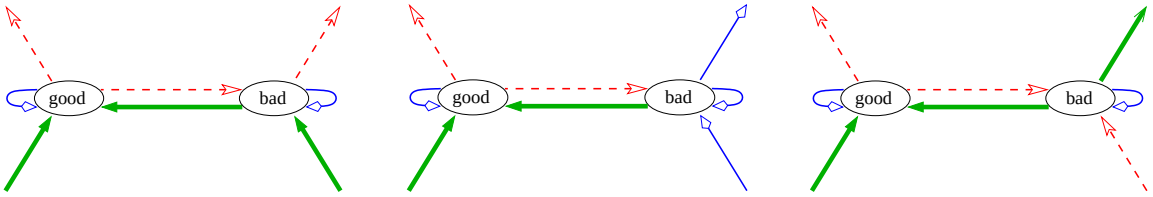
flipping each bit with probability $1/n$. $h'$ is accepted as a viable neighbor only if $|h'| \leq q$. `USelect` picks uniformly at random among the elements of the set passed as argument. The parameter $\epsilon_s$ governs the approximation within which the empirical performance of a hypothesis will be computed (compared to its true performance) and such an approximation will be correct except with probability at most $\delta_s$. The value of $\delta_s$ is selected in such a way, so that eventually with a union bound (we will see the details in the proofs that are available in Section 4.4) one can provide the guarantee that we want: i.e., the evolution succeeds in every phase with *overall probability* at least $1 - \delta$. This is why these values of $\epsilon_s$ and $\delta_s$ are passed as parameters in the computation of the empirical performance of the hypotheses $h$ and $h'$ in lines 15 and 16 (instead of the global parameters $\varepsilon$ and $\delta$).

## 3. Foundations for Evolvability

For a current hypothesis $h$, of particular interest will be the hypotheses that have: Hamming distance 1, or Hamming distance 2 and same size, with respect to $h$. The set of hypotheses that is obtained from $h$ by *adding* (*removing*) a variable is denoted by $N^+$ (respectively, $N^-$). The set of hypotheses that is obtained from $h$ by *swapping* a variable with another one is denoted by $N^{\pm}$. As an example, let $h = x_1 \wedge x_2$, and $n = 3$. Then, $N^- = \{x_1, x_2\}$, $N^+ = \{x_1 \wedge x_2 \wedge x_3\}$, and $N^{\pm} = \{x_3 \wedge x_2, x_1 \wedge x_3\}$.

Even if we have to take into account mutations of $h$ that are wilder than those described by $N^+, N^-$ and $N^{\pm}$, nevertheless, such simple mutations are important for proving Theorem 3, Lemmas 1 and 2, as well as for some arguments related to the convergence of the (1+1)

EA. Figure 3 presents the sign of the difference $\Delta$ for such mutations that give rise to a hypothesis in $N^+$, $N^-$ or $N^\pm$ as explained by the three quantities below.




(a) $U < 1/2$ (b) $U = 1/2$ (c) $U > 1/2$

Figure 3: Arrows pointing towards the nodes indicate addition of one variable and arrows pointing away from a node indicate removal of one variable. This is consistent with arrows indicating swapping a pair of variables. Let $\Delta = \mathrm{Perf}_{\mathcal{B}_n}(h', c) - \mathrm{Perf}_{\mathcal{B}_n}(h, c)$. Thick solid lines indicate $\Delta > 0$. Simple lines indicate $\Delta = 0$. Dashed lines indicate $\Delta < 0$. Figure 3a holds when $U < 1/2$; Figure 3b when $U = 1/2$; Figure 3c when $U > 1/2$.

- Comparing $h' \in N^+$ with $h$. We introduce a variable $z$ in the hypothesis $h$. If $z$ is good, $\Delta = 2p^{|h|}(1-p) > 0$. If $z$ is bad, $\Delta = 2p^{|h|}(1-2p^u)(1-p)$.
- Comparing $h' \in N^-$ with $h$. We remove a variable $z$ from the hypothesis $h$. If $z$ is good, $\Delta = -2p^{|h|-1}(1-p) < 0$. If $z$ is bad, $\Delta = -2p^{|h|-1}(1-2p^u)(1-p)$.
- Comparing $h' \in N^\pm$ with $h$. Replacing a good with a bad variable gives $\Delta = -4p^{|h|+u}(1-p)$. Replacing a good with a good, or a bad with a bad variable gives $\Delta = 0$. Replacing a bad with a good variable gives $\Delta = 4p^{|h|+u-1}(1-p)$.

While the sign of an arrow in Figure 3 may be fully determined, it is the value of the tolerance $t$ that defines the sets Bene and Neut that guide the search. A critical quantity in the above calculations is $\mathcal{A}(u) = |1-2p^u|$, $u \in \{0, \ldots, n\}$; its minimum non-zero value of $\mathcal{A}(u)$, $\min_{\neq 0}\{\mathcal{A}(u)\}$, is discussed by Diochnos (2016) for every $p \in (0,1)$. In our context, for $p \in (0, 1/3]$ we have $\min_{\neq 0}\{\mathcal{A}(u)\} = 1-2p \geq 1/3$, while for $p = 1/2$ we have $\min_{\neq 0}\{\mathcal{A}(u)\} = 1/2$.

**Theorem 3** (Diochnos, 2016). *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ product distribution with parameter $p$. The best $q$-approximation of a target $c$ is $c$ if $|c| \leq q$, or any hypothesis formed by $q$ good variables if $|c| > q$.*

Lemmas 1 and 2 below are taken from the work of Diochnos (2016), where for completeness, we give their short proofs. Lemma 1 is relevant in our context only under $\mathcal{U}_n$, where $\vartheta = 1$. While Algorithm 2 does not mention $\vartheta$, it is however taken into account under $\mathcal{U}_n$ in Lemmas 4, 11 and 12, when computing tolerance and sample size.

**Lemma 1** (Medium Targets). *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ distribution, $h$ a hypothesis and $c$ be the target. Then, $q \geq \log_{\frac{1}{p}}\left(\frac{3}{\varepsilon}\right), \vartheta \geq 0, |h| = q < |c| \leq q + \vartheta$, all variables in $h$ are good $\Rightarrow \mathit{Perf}_{\mathcal{B}_n}(h, c) > 1 - 2\varepsilon/3$.*

*Proof.* The setup of the lemma implies $m = q$, $r = 0$, $u \leq \vartheta$. Using (6) we have: $\text{Perf}_{\mathcal{B}_n}(h,c) = 1 - 2p^{m+r} - 2p^{m+u} + 4p^{m+r+u} = 1 - 2p^q + 2p^{q+\vartheta} > 1 - 2p^q$. It follows that $\text{Perf}_{\mathcal{B}_n}(h,c) > 1 - 2p^q \geq 1 - 2\varepsilon/3$. □

**Lemma 2** (Long Targets)**.** *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ distribution, $h$ a hypothesis and $c$ the target. Then, $q \geq \log_{\frac{1}{p}}\left(\frac{3}{\varepsilon}\right)$, $\vartheta \geq \log_{\frac{1}{p}}(2p)$, $|h| \geq q$, $|c| > q + \vartheta \Rightarrow Perf_{\mathcal{B}_n}(h,c) > 1 - \varepsilon$.*

*Proof.* The setup of the lemma implies $m + r \geq q$, and $m + u > q + \vartheta$. Using (6), $\text{Perf}_{\mathcal{B}_n}(h,c) > 1 - 2p^{m+r} - 2p^{m+u} \geq 1 - 2p^q - 2p^{q+\vartheta+1} = 1 - 2p^q\left(1 + p^{1+\vartheta}\right)$. It follows that $\text{Perf}_{\mathcal{B}_n}(h,c) > 1 - 2p^q\left(1 + p^{1+\vartheta}\right) \geq 1 - 2 \cdot \frac{\varepsilon}{3} \cdot \left(1 + p^{\log_{\frac{1}{p}}(2)}\right) = 1 - \varepsilon$. □

## 4. Analysis of Our Modified (1+1) Evolutionary Algorithm

We start with some coarse characterizations under $\mathcal{U}_n$ for the three sets into which the hypothesis space $\mathcal{H} = \mathcal{H}_{<1/2} \cup \mathcal{H}_{1/2} \cup \mathcal{H}_{>1/2}$ is partitioned. Lemma 3 further refines the hypotheses in $\mathcal{H}_{<1/2}$ and expresses the fact that among such hypotheses, the larger the size of the hypothesis, the better its performance. The lemmas for the relevant fitness levels for the case where $p \in (0, 1/3]$ are presented in a separate section on their own.

### 4.1 Fitness Levels when $p = 1/2$

**Lemma 3** (More Variables in the Hypothesis is Better in $\mathcal{H}_{<1/2}$ under $\mathcal{U}_n$)**.** *Let $h, h' \in \mathcal{H}_{<1/2}$ such that $|h'| < |h| \leq q$. Then, under the uniform distribution $\mathcal{U}_n$, $Perf_{\mathcal{U}_n}(h,c) \geq Perf_{\mathcal{U}_n}(h',c) + 2^{1-2q}$.*

*Proof.* We will prove the lemma by distinguishing cases on the size of the target. The technique is identical to the case where $p = 1/3$. Let $|h| = \lambda + |h'|$, for $\lambda \geq 1$.

**Case** $|c| \leq q + 1$**.** By (6), we have $\text{Perf}_{\mathcal{U}_n}(h,c) = 1 - 2^{1-|c|} - 2^{1-|h|} + 2^{2-|c|-r}$. Since $|h| = \lambda + |h'|$ and moreover the number $r$ of redundant variables in $h$ can be $r \in \{0, \ldots, q\}$, it follows that

$$\text{Perf}_{\mathcal{U}_n}(h,c) \geq 1 - 2^{1-|c|} - 2^{1-\lambda-|h'|} + 2^{2-|c|-q}. \tag{7}$$

On the other hand, by (6), letting $u'$ be the number of good undiscovered variables in $h'$, we have $\text{Perf}_{\mathcal{U}_n}(h',c) = 1 - 2^{1-|c|} - 2^{1-|h'|} + 2^{2-|h'|-u'}$. Since $h' \in \mathcal{H}_{<1/2} \Rightarrow u' \geq 2$ it follows that

$$\begin{aligned} \text{Perf}_{\mathcal{U}_n}\left(h',c\right) &\leq 1 - 2^{1-|c|} - 2^{1-|h'|} + 2^{-|h'|} \\ &= 1 - 2^{1-|c|} - 2^{-|h'|}. \end{aligned} \tag{8}$$

Thus, by (7) and (8) it follows that

$$\begin{aligned} \Delta &= \text{Perf}_{\mathcal{U}_n}(h,c) - \text{Perf}_{\mathcal{U}_n}\left(h',c\right) \\ &\geq 2^{-|h'|} \cdot (1 - 2^{1-\lambda}) + 2^{2-|c|-q} \\ &\geq 2^{1-2q}, \end{aligned} \tag{9}$$

where the last inequality is obtained since $\lambda \geq 1$ and $|c| \leq q + 1$.

**Case** $|c| \geq q+2$**.** For $h$, (7) continues to hold. On the other hand, letting $r'$ be the number of (bad) redundant variables in $h'$, we have $\mathrm{Perf}_{\mathcal{U}_n}(h', c) = 1 - 2^{1-|c|} - 2^{1-|h'|} + 2^{2-|c|-r'}$, and thus

$$\mathrm{Perf}_{\mathcal{U}_n}\left(h', c\right) \leq 1 - 2^{1-|c|} - 2^{1-|h'|} + 2^{2-|c|}\,. \tag{10}$$

Hence, by (7) and (10) we have

$$\begin{aligned}
\Delta &= \mathrm{Perf}_{\mathcal{U}_n}(h, c) - \mathrm{Perf}_{\mathcal{U}_n}\left(h', c\right) \\
&\geq 2^{1-|h'|} \cdot (1 - 2^{-\lambda}) - 2^{2-|c|} \cdot (1 - 2^{-q}) \\
&\geq 2^{1-(q-1)} \cdot \frac{1}{2} - 2^{2-|c|} \\
&\geq 2^{1-q} - 2^{-q} \\
&= 2^{-q}\,.
\end{aligned}$$

The lemma follows by observing that since $q \geq 1$, we have that $1 \geq 2^{1-q} \Rightarrow 2^{-q} \geq 2^{1-2q}$. □

**Lemma 4** ($\mathcal{H}_{1/2} \not\rightsquigarrow \mathcal{H}_{<1/2}$)**.** *Under $\mathcal{U}_n$, let $h \in \mathcal{H}_{1/2}$ and $h' \in \mathcal{H}_{<1/2}$. Then, $Perf_{\mathcal{U}_n}(h, c) \geq Perf_{\mathcal{U}_n}(h', c) + 2^{-q}$.*

*Proof.* First, $h \in \mathcal{H}_{1/2} \Rightarrow u = 1$. Then, by (6),

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{U}_n}(h, c) &= 1 - 2^{1-|h|} - 2^{1-|c|} + 2^{2-|h|-1} \\
&= 1 - 2^{1-|c|}\,.
\end{aligned}$$

On the other hand, $h' \in \mathcal{H}_{<1/2} \Rightarrow u' \geq 2$. Then, by (6),

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{U}_n}\left(h', c\right) &= 1 - 2^{1-|h'|} - 2^{1-|c|} + 2^{2-|h'|-u'} \\
&\leq 1 - 2^{1-|h'|} - 2^{1-|c|} + 2^{2-|h'|-2} \\
&= 1 - 2^{-|h'|} - 2^{1-|c|} \\
&\leq 1 - 2^{-q} - 2^{1-|c|}\,.
\end{aligned}$$

It follows that $\mathrm{Perf}_{\mathcal{U}_n}(h, c) - \mathrm{Perf}_{\mathcal{U}_n}(h', c) \geq 2^{-q}$. □

**Lemma 5** ($\mathcal{H}_{>1/2} \not\rightsquigarrow \mathcal{H}_{1/2}$)**.** *Under $\mathcal{U}_n$, let $h \in \mathcal{H}_{>1/2}$ and $h' \in \mathcal{H}_{1/2}$. Then, $Perf_{\mathcal{U}_n}(h, c) \geq Perf_{\mathcal{U}_n}(h', c) + 2^{1-q}$.*

*Proof.* First, $h \in \mathcal{H}_{>1/2} \Rightarrow u = 0$. Then, by (6),

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{U}_n}(h, c) &= 1 - 2^{1-|h|} - 2^{1-|c|} + 2^{2-|h|-u} \\
&= 1 - 2^{1-|h|} - 2^{1-|c|} + 2^{2-|h|} \\
&= 1 + 2^{1-|h|} - 2^{1-|c|} \\
&\geq 1 + 2^{1-q} - 2^{1-|c|}\,.
\end{aligned}$$

On the other hand, $h' \in \mathcal{H}_{1/2} \Rightarrow u' = 1$. Then, by (6),

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{U}_n}\left(h', c\right) &= 1 - 2^{1-|h'|} - 2^{1-|c|} + 2^{2-|h'|-1} \\
&= 1 - 2^{1-|c|}\,.
\end{aligned}$$

It follows that $\mathrm{Perf}_{\mathcal{U}_n}(h, c) - \mathrm{Perf}_{\mathcal{U}_n}(h', c) \geq 2^{1-q}$. □

**Lemma 6** ($\mathcal{H}_{>1/2} \not\rightsquigarrow \mathcal{H}_{<1/2}$ under $\mathcal{U}_n$)**.** *Under the uniform distribution $\mathcal{U}_n$, let $h \in \mathcal{H}_{>1/2}$ and $h' \in \mathcal{H}_{<1/2}$. Then, $Perf_{\mathcal{U}_n}(h,c) \geq Perf_{\mathcal{U}_n}(h',c) + 3 \cdot 2^{-q}$.*

*Proof.* First, $h \in \mathcal{H}_{>1/2} \Rightarrow u = 0$. Then, by (6),

$$\begin{aligned} \mathrm{Perf}_{\mathcal{U}_n}(h,c) &= 1 - 2^{1-|h|} - 2^{1-|c|} + 2^{2-|h|-u} \\ &= 1 - 2^{1-|h|} - 2^{1-|c|} + 2^{2-|h|} \\ &= 1 + 2^{1-|h|} - 2^{1-|c|} \\ &\geq 1 + 2^{1-q} - 2^{1-|c|}\,. \end{aligned}$$

On the other hand, $h' \in \mathcal{H}_{<1/2} \Rightarrow u' \geq 2$. Then, by (6),

$$\begin{aligned} \mathrm{Perf}_{\mathcal{U}_n}\left(h',c\right) &= 1 - 2^{1-|h'|} - 2^{1-|c|} + 2^{2-|h'|-u'} \\ &\leq 1 - 2^{1-|h'|} - 2^{1-|c|} + 2^{2-|h'|-2} \\ &= 1 - 2^{-|h'|} - 2^{1-|c|} \\ &\leq 1 - 2^{-q} - 2^{1-|c|}\,. \end{aligned}$$

Thus, $\mathrm{Perf}_{\mathcal{U}_n}(h,c) - \mathrm{Perf}_{\mathcal{U}_n}(h',c) \geq 3 \cdot 2^{-q}$. □

### 4.2 Fitness Levels when $p \in (0, 1/3]$

**Lemma 7** (More Variables in the Hypothesis is Better in $\mathcal{H}_{<1/2}$ when $0 < p < \frac{1}{3}$)**.** *Let $h, h' \in \mathcal{H}_{<1/2}$ such that $|h'| < |h| \leq q$. Then, under a Bernoulli$(p)^n$ distribution $\mathcal{B}_n$ with $p \in (0,1/3)$, $Perf_{\mathcal{B}_n}(h,c) \geq Perf_{\mathcal{B}_n}(h',c) + 2p^{q-1} \cdot (1-3p)$.*

*Proof.* Let the target be

$$c = \bigwedge_{i=1}^{m_1} x_i \wedge \bigwedge_{i=m_1+1}^{m} x_i \wedge \bigwedge_{k=1}^{u_1} y_k \wedge \bigwedge_{k=u_1+1}^{u} y_k\,.$$

Furthermore let,

$$\begin{cases} h = \bigwedge_{i=1}^{m_1} x_i \wedge \bigwedge_{i=m_1+1}^{m} x_i \wedge \bigwedge_{\ell=1}^{r_1} w_\ell \wedge \bigwedge_{\ell=r_1+1}^{r} w_\ell \\ h' = \bigwedge_{i=1}^{m_1} x_i \wedge \bigwedge_{k=1}^{u_1} y_k \wedge \bigwedge_{\ell=1}^{r_1} w_\ell \wedge \bigwedge_{j=1}^{r_3} z_j \end{cases}$$

be the two short hypotheses, such that $|h| = |h'| + \lambda$ for $\lambda \geq 1$ and moreover, $U = \prod_{k=1}^{u} p_{y_k} = p^u < 1/2$ and $U' = \left(\prod_{i=m_1+1}^{m} p_{x_i}\right) \cdot \left(\prod_{k=u_1+1}^{u} p_{y_k}\right) = p^{m_2} \cdot p^{u_2} < 1/2$, where $m_2 = |\{x_{m_1+1}, \ldots, x_m\}|$, $u_2 = |\{y_{u_1+1}, \ldots, y_u\}|$ and $r_2 = |\{w_{r_1+1}, \ldots, w_r\}|$.

By construction we have $|h| = m_1 + m_2 + r_1 + r_2 = \lambda + m_1 + u_1 + r_1 + r_3 = \lambda + |h'|$. In other words, it holds

$$m_2 + r_2 = u_1 + r_3 + \lambda\,. \tag{11}$$

For the difference $\Delta$ in performance we have

$$\begin{aligned}
\Delta &= \mathrm{Perf}_{\mathcal{B}_n}(h,c) - \mathrm{Perf}_{\mathcal{B}_n}\left(h',c\right) \\
&= 2p^{m_1+u_1+r_1+r_3} + 2p^{m_1+m_2+u_1+u_2} - 4p^{m_1+m_2+u_1+u_2+r_1+r_3} \\
&\quad -2p^{m_1+m_2+r_1+r_2} - 2p^{m_1+m_2+u_1+u_2} + 4p^{m_1+m_2+u_1+u_2+r_1+r_2} \\
&= 2p^{m_1+r_1}\cdot\left(p^{u_1+r_3} - p^{m_2+r_2}\right) \\
&\quad +2p^{m_1+r_1}\cdot\left(2p^{m_2+u_1+u_2+r_2} - 2p^{m_2+u_1+u_2+r_3}\right) \\
&= 2p^{m_1+r_1}\cdot\left(p^{u_1+r_3} - p^{u_1+r_3+1}\right) \\
&\quad +2p^{m_1+r_1}\cdot\left(2p^{m_2+u_1+u_2+r_2} - 2p^{m_2+u_1+u_2+r_3}\right) \\
&= 2p^{m_1+r_1+u_1+r_3} - 2p^{m_1+r_1+u_1+r_3}p \\
&\quad +4p^{m_1+m_2+r_1+r_2+u_1+u_2} - 4p^{m_1+m_2+r_1+r_3+u_1+u_2} \\
&= 2p^{m_1+r_1+u_1+r_3}\cdot(1-p) + 2p^{m_1+r_1+u_1+r_3}\cdot\left(2p^{m_2+r_2+u_2-r_3} - 2p^{m_2+u_2}\right) \\
&\geq 2p^{|h'|}\cdot\left(1-p+2p^{\lambda+u_1+u_2} - 2p\right) \qquad (12) \\
&> 2p^{|h'|}\cdot(1-3p) \\
&\geq 2p^{q-1}\cdot(1-3p)\,,
\end{aligned}$$

where (12) follows by (11) and the fact that $p^{m_2+u_2} = U' < 1/2 \Rightarrow m_2 + u_2 \geq 1$. The claim follows. □

**Lemma 8** (More Variables in the Hypothesis is Better in $\mathcal{H}_{<1/2}$ when $p = \frac{1}{3}$)**.** *Let $h, h' \in \mathcal{H}_{<1/2}$ such that $|h'| < |h| \leq q$. Then, under a Bernoulli$(p)^n$ distribution $\mathcal{B}_n$ with $p = 1/3$, $Perf_{\mathcal{B}_n}(h,c) \geq Perf_{\mathcal{B}_n}(h',c) + \frac{4}{3}\cdot 3^{-2q}$.*

*Proof.* We will prove the lemma by distinguishing cases on the size of the target. Let $|h| = \lambda + |h'|$, for $\lambda \geq 1$.

**Case** $|c| \leq q+1$**.** By (6), we have $\mathrm{Perf}_{\mathcal{B}_n}(h,c) = 1 - 2\cdot 3^{-|c|} - 2\cdot 3^{-\lambda-|h'|} + 4\cdot 3^{-|c|-r}$. Since the number $r$ of redundant variables in $h$ can be $r \in \{0,\ldots,q\}$, we have,

$$\mathrm{Perf}_{\mathcal{B}_n}(h,c) \geq 1 - 2\cdot 3^{-|c|} - 2\cdot 3^{-\lambda-|h'|} + 4\cdot 3^{-|c|-q}\,. \qquad (13)$$

On the other hand, by (6), letting $u'$ be the number of good undiscovered variables in $h'$, we have $\mathrm{Perf}_{\mathcal{B}_n}(h',c) = 1 - 2\cdot 3^{-|c|} - 2\cdot 3^{-|h'|} + 4\cdot 3^{-|h'|-u'}$. Since $h' \in \mathcal{H}_{<1/2} \Rightarrow u' \geq 1$ it follows that

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{B}_n}\left(h',c\right) &\leq 1 - 2\cdot 3^{-|c|} - 2\cdot 3^{-|h'|} + 4\cdot 3^{-|h'|-1} \\
&= 1 - 2\cdot 3^{-|c|} - 2\cdot 3^{-|h'|-1}\,. \qquad (14)
\end{aligned}$$

Thus, by (13) and (14) it follows that

$$\begin{aligned}
\Delta &= \mathrm{Perf}_{\mathcal{B}_n}(h,c) - \mathrm{Perf}_{\mathcal{B}_n}\left(h',c\right) \\
&\geq 2\cdot 3^{-|h'|-1} - 2\cdot 3^{-|h'|-\lambda} + 4\cdot 3^{-|c|-q} \\
&\geq 4\cdot 3^{-2q-1}\,,
\end{aligned}$$

where the last inequality is obtained since $|c| \leq q+1$ and we also used the fact that $\lambda \geq 1$.

**Case** $|c| \geq q+2$**.** For $h$, (13) continues to hold. On the other hand, letting $r'$ be the number of (bad) redundant variables in $h'$, we have $\mathrm{Perf}_{\mathcal{B}_n}(h', c) = 1 - 2 \cdot 3^{-|c|} - 2 \cdot 3^{-|h'|} + 4 \cdot 3^{-|c|-r'}$, and thus

$$\mathrm{Perf}_{\mathcal{B}_n}\left(h', c\right) \leq 1 - 2 \cdot 3^{-|c|} - 2 \cdot 3^{-|h'|} + 4 \cdot 3^{-|c|}\,. \tag{15}$$

Hence, by (13) and (15) we have

$$\begin{aligned}
\Delta &= \mathrm{Perf}_{\mathcal{B}_n}(h, c) - \mathrm{Perf}_{\mathcal{B}_n}\left(h', c\right) \\
&\geq 2 \cdot 3^{-|h'|} - 2 \cdot 3^{-|h'|-\lambda} + 4 \cdot 3^{-|c|-q} - 4 \cdot 3^{-|c|} \\
&> 2 \cdot 3^{-|h'|} \cdot \left(1 - 3^{-\lambda}\right) - 4 \cdot 3^{-|c|} \\
&\geq 2 \cdot 3^{1-q} \cdot \frac{2}{3} - 4 \cdot 3^{-q-2} \\
&= \frac{32}{9} \cdot 3^{-q}\,.
\end{aligned}$$

Since $q \geq 1$, the lemma follows by observing that $\frac{32}{9} \cdot 3^{-q} > \frac{4}{3} \cdot 3^{-q} \geq \frac{4}{3} \cdot 3^{-2q}$. □

**Lemma 9** ($\mathcal{H}_{>1/2} \not\rightsquigarrow \mathcal{H}_{<1/2}$ under $\mathcal{B}_n$ with $p \in (0, 1/3]$)**.** *Under a Bernoulli*$(p)^n$ *distribution* $\mathcal{B}_n$ *with parameter* $p \in (0, 1/3]$*, let* $h \in \mathcal{H}_{>1/2}$ *and* $h' \in \mathcal{H}_{<1/2}$ *such that* $|h| \leq q$ *and* $|h'| \leq q$*. Then,* $Perf_{\mathcal{B}_n}(h, c) \geq Perf_{\mathcal{B}_n}(h', c) + \frac{8}{3} \cdot p^q$*.*

*Proof.* First, $h \in \mathcal{H}_{>1/2} \Rightarrow u = 0$. Then, by (6),

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{B}_n}(h, c) &= 1 - 2p^{|h|} - 2p^{|c|} + 4p^{|h|+u} \\
&= 1 - 2p^{|h|} - 2p^{|c|} + 4p^{|h|+0} \\
&= 1 + 2p^{|h|} - 2p^{|c|}\,.
\end{aligned}$$

Before we proceed, note that for $p \in (0, 1/3]$, we have $1 - 2p^u \geq 1 - 2p \geq 1 - 2/3 = 1/3$ for any integer $u \geq 1$. Then, since $h' \in \mathcal{H}_{<1/2}$ we have $u' \geq 1$. As a result, by (6),

$$\begin{aligned}
\mathrm{Perf}_{\mathcal{B}_n}\left(h', c\right) &= 1 - 2p^{|h'|} - 2p^{|c|} + 4p^{|h'|+u'} \\
&= 1 - 2p^{|c|} - 2p^{|h'|}\left(1 - 2p^{u'}\right) \\
&\leq 1 - 2p^{|c|} - \frac{2}{3} \cdot p^{|h'|}\,.
\end{aligned}$$

It follows that $\mathrm{Perf}_{\mathcal{B}_n}(h, c) - \mathrm{Perf}_{\mathcal{B}_n}(h', c) \geq 2p^{|h|} + \frac{2}{3} \cdot p^{|h'|} \geq 2p^q + \frac{2}{3} \cdot p^q = \frac{8}{3} \cdot p^q$. □

### 4.3 Convergence

We start with the lemmas that signify the different phases of the algorithm in every case where $p \in (0, 1/3] \cup \{1/2\}$. We use the terms *generalization* and *specialization* as in Tom Mitchell's framework of *version spaces* (Mitchell, 1997). That is, a Boolean function $f$ is a *generalization* (resp., *specialization*) of a Boolean function $f'$ iff the set of satisfying truth assignments for $f$ is a *superset* (resp., *subset*) of the set of satisfying truth assignments for $f'$.

**Lemma 10** (Long Targets)**.** *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ distribution with parameter $p \in \mathbb{R}_{alg}$ such that $p \in (0, 1/3] \cup \{1/2\}$. Starting with a short hypothesis $h_0$, the (1+1) EA, within $\lceil 16enq/\delta \rceil$ generations, assuming that the performance of each hypothesis generated is estimated within $\epsilon_s = t/2$ of its true value, with probability at least $1 - \delta/16$, will evolve a hypothesis $h$ such that, either $h \in \mathcal{H}_{1/2} \cup \mathcal{H}_{>1/2}$, or it is the case that $h \in \mathcal{H}_{<1/2}$ and $|h| = q$.*

*Proof.* Suffices to prove the lemma for an initial hypothesis $h_0 \in \mathcal{H}_{<1/2}$, otherwise the statement is trivial as $h_0 \in \mathcal{H}_{1/2} \cup \mathcal{H}_{>1/2}$. As long as $h \in \mathcal{H}_{<1/2}$ and $|h| < q$, the probability of adding at least one good or bad variable to the hypothesis is lower bounded by the probability of the event of not touching the variables that appear in $h$ and introducing precisely one variable. In other words, for this probability it holds $(1 - 1/n)^{n-1} \cdot \frac{1}{n} \geq \frac{1}{e} \cdot \frac{1}{n}$. Thus, the expected time to introduce at least one variable is at most $en$. Conditioning on $h \in \mathcal{H}_{<1/2}$ throughout, the expected time to form a hypothesis of size precisely $q$ is at most $enq$. We now apply Markov's inequality with failure probability $\delta/16$. □

**Lemma 11** (Specialization under $p \in (0, 1/3]$ or Best Approximation under $\mathcal{U}_n$)**.** *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ distribution with parameter $p \in \mathbb{R}_{alg}$; $p \in (0, 1/3] \cup \{1/2\}$. Unless the target is long, starting with a hypothesis $h_0$ such that $h_0 \in \mathcal{H}_{<1/2}$ and $|h_0| = q$, the (1+1) EA, within $\lceil 16en^2q/\delta \rceil$ generations, assuming that the performance of each hypothesis generated is estimated within $\epsilon_s = t/2$ of its true value, with probability at least $1 - \delta/16$, will evolve a hypothesis $h \in \mathcal{H}_{1/2} \cup \mathcal{H}_{>1/2}$.*

*Proof.* Due to Lemmas 7, 8 and 3, as long as $h \in \mathcal{H}_{<1/2}$, any mutations that form a hypothesis $h' \in \mathcal{H}_{<1/2}$ such that $|h'| < |h|$ will cause a noticeable decrease in performance by the selection of tolerance. On the other hand, swapping a bad with a good variable provides a noticeable increase in performance, again due to the selection of tolerance, and occurs with probability at least $(1 - 1/n)^{n-2} \cdot \frac{1}{n} \cdot \frac{1}{n} > \frac{1}{e} \cdot \frac{1}{n^2}$. Conditioning on $h \in \mathcal{H}_{<1/2}$, up to $q$ such swaps will occur within expected time not more than $en^2q$. However, for short and medium targets, this implies enough swaps that lead to $u = 0$ when $p \in (0, 1/3]$ or $u = 1$ when $p = 1/2$ and thus in either case a hypothesis $h$ is formed such that $h \in \mathcal{H}_{1/2} \cup \mathcal{H}_{>1/2}$. We now apply Markov's inequality with failure probability $\delta/16$.

Note that the final hypothesis $h$ belongs to $\mathcal{H}_{1/2}$ only under $\mathcal{U}_n$; if it is the case that evolution takes place under a Bernoulli$(p)^n$ distribution with parameter $p \in (0, 1/3]$, then $h$ belongs to $\mathcal{H}_{>1/2}$, which in fact implies that $u = 0$. □

**Lemma 12** (Maintain Best $q$-Approximation of Medium Targets or Create a Specialization of Short Targets under $\mathcal{U}_n$)**.** *Under the uniform distribution $\mathcal{U}_n$, let $h_0 \in \mathcal{H}_{1/2}$ such that $|h_0| \leq q$. The (1+1) EA, within $\lceil 16en^2/\delta \rceil$ generations, assuming that the performance of each hypothesis generated is estimated within $\epsilon_s = t/2$ of its true value, with probability at least $1 - \delta/16$, will, either maintain a best $q$-approximation for a target of size $q + 1$, or evolve a hypothesis $h \in \mathcal{H}_{>1/2}$.*

*Proof.* We have $u = 1$, corresponding to Figure 3b.

First of all, if the target has size $|c| = q + 1$, then, since $u = 1$, we have $|h| = m = q$. That is, $h$ is a best $q$-approximation of $c$. Due to Lemma 4, $h$ is noticeably better (by

an amount of at least $2^{-q}$) compared to any other hypothesis of size at most $q$ that is not a best $q$-approximation (as for any such other hypothesis $h'$ it holds $h' \in \mathcal{H}_{<1/2}$). Thus, the formation is stable and $h$ can only mutate to other hypotheses that are also best $q$-approximations.

In the other case, $|c| \leq q$. (Targets with size $|c| > q+1$ imply $u \geq 2 \Rightarrow h \in \mathcal{H}_{<1/2}$ contradicting that $h \in \mathcal{H}_{1/2}$.) Neutral mutations do not affect the number of good variables that already appear in $h$. Also, any beneficial mutation results in the introduction of the last good variable that is missing from the hypothesis. Due to Lemma 5 introducing the last missing good variable results in noticeable increase in performance. Furthermore, by Lemma 4, any hypothesis that has fewer good variables than $h$ results in a noticeable decrease in performance. Thus, neutral mutations in this phase will generate hypotheses with sizes in $\{m, m+1, \ldots, q\} = \{|c|-1, |c|, \ldots, q\}$ depending on the number of present bad variables in these hypotheses. In every generation the last good variable is introduced into the hypothesis due to either a beneficial swap or a beneficial addition of the last good variable. When $|h| < q$, the last good variable is added with probability at least $(1-1/n)^{n-1} \cdot \frac{1}{n} \geq \frac{1}{en}$. When $|h| = q$ the probability of a beneficial swap is at least $(1-1/n)^{n-2} \cdot \frac{1}{n} \cdot \frac{1}{n} \geq \frac{1}{en^2}$. Hence, regardless of the size of $h$, the probability that the last good variable is introduced into the hypothesis within one generation, is at least $\frac{1}{en^2}$. We now apply Markov's inequality with failure probability $\delta/16$. □

**Lemma 13** (Identification of Short Targets)**.** *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ distribution with parameter $p \in \mathbb{R}_{alg}$ such that $p \in (0, 1/3] \cup \{1/2\}$. For an initial hypothesis $h_0 \in \mathcal{H}_{>1/2}$ such that $|h_0| \leq q$, the (1+1) EA, within $\lceil 16enq/\delta \rceil$ generations, assuming that the performance of each hypothesis generated is estimated within $\epsilon_s = t/2$ of its true value, with probability at least $1 - \delta/16$, will evolve to the target $c$.*

*Proof.* $h_0 \in \mathcal{H}_{>1/2} \Rightarrow u = 0$, corresponding to Figure 3c. Furthermore, $m = |c| \leq q$. Any hypothesis that is missing $u \geq 1$ variables has noticeably smaller performance compared to any hypothesis that is a specialization of the target ($u = 0$). Under $\mathcal{U}_n$ Lemmas 5 and 6 provide the performance gap between such hypotheses, while for Bernoulli$(p)^n$ distributions $\mathcal{B}_n$ with $p \in (0, 1/3]$ the gap in performance is provided by Lemma 9.

If the starting hypothesis $h_0$ contains redundant bad variables, then beneficial mutations are those that remove one or more of those in one step. Such a beneficial removal of one bad variable will occur within one generation with probability at least $(1-1/n)^{n-1} \cdot \frac{1}{n} \geq \frac{1}{en}$ and will be identified as such. Since $h_0$ contains not more than $q$ bad variables, by linearity of expectation, it follows that within $enq$ generations all the redundant bad variables are expected to be removed from $h_0$ thus leading to the target $c$. In this formation the only neutral mutation is the target itself. We now apply Markov's inequality with failure probability $\delta/16$. □

### 4.4 Complexity

The complexity analysis has to be performed for three different cases; once for the uniform distribution ($p = 1/2$), once when $p = 1/3$, and once when $p \in (0, 1/3)$. Below we present a unifying theorem for all these three cases.

**Theorem 4.** *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ distribution with parameter $p \in \mathbb{R}_{alg}$ such that $p \in (0, 1/3] \cup \{1/2\}$. Let $q = \lceil \log_{1/p}(3/\varepsilon) \rceil$. Then, using the hypothesis class $\mathcal{H} = \mathcal{C}_n^{\leq q}$, starting from any initial hypothesis, the (1+1) EA will evolve a hypothesis $h$ such that $\mathbf{Pr}\left(Perf_{\mathcal{B}_n}(h, c) > 1 - \varepsilon\right) \geq 1 - \delta$, in $\mathcal{O}\left(n^2 q/\delta\right)$ generations. The total sample size is $\widetilde{\mathcal{O}}\left(\frac{n^2 q}{\delta \cdot \varepsilon^4}\right)$ when $p = 1/3$ or $p = 1/2$, and $\widetilde{\mathcal{O}}\left(\frac{n^2 q}{\delta \cdot \varepsilon^2 \cdot (\min\{4p\varepsilon/9, 1-3p\})^2}\right)$ when $0 < p < 1/3$.*

The proofs of these three different cases are similar. One has to use the appropriate lower bounds for the tolerance in every case. Evolution under the uniform distribution appears to be the more involved among the three of these cases, since it allows hypotheses to visit all three diagrams in Figure 3. For this reason, we start with the presentation of the complexity analysis for the uniform distribution and in sequence we examine the complexity analysis for the cases where $p \in (0, 1/3)$ or $p = 1/3$.

#### 4.4.1 Complexity Analysis under the Uniform Distribution

**Theorem 5** (Evolution in $\mathcal{C}_n^{\leq q}$ under $\mathcal{U}_n$)**.** *Let $q = \lceil \log_2(3/\varepsilon) \rceil$. Under the uniform distribution $\mathcal{U}_n$, using the hypothesis class $\mathcal{H} = \mathcal{C}_n^{\leq q}$, starting from any initial hypothesis, the (1+1) EA will evolve a hypothesis $h$ such that $\mathbf{Pr}\left(Perf_{\mathcal{U}_n}(h, c) > 1 - \varepsilon\right) \geq 1 - \delta$, in $\mathcal{O}\left(n^2 q/\delta\right)$ generations using total sample size $\widetilde{\mathcal{O}}\left(n^2 q/(\delta\varepsilon^4)\right)$.*

*Proof.* Note that long targets are taken care of by Lemma 10 as soon as a hypothesis of size $q$ has been formed. The reason is that any approximation of a long target belongs to $\mathcal{H}_{<1/2}$ and due to Lemma 3 all shorter hypotheses have performance that will be noticeably smaller by the selection of tolerance. Thus, below we will only discuss about short and medium targets.

**Phase 1.** In the first phase of the evolution (Lemma 10), as long as $h \in \mathcal{H}_{<1/2}$, increasing the size of $h$ results in increase in performance by an amount of at least $2^{1-2q}$ due to Lemma 3. Lemma 10 serves its purpose when either $h \in \mathcal{H}_{1/2}$ (leading to phase 3), or $h \in \mathcal{H}_{>1/2}$ (leading to phase 4), or it is still the case that $h \in \mathcal{H}_{<1/2}$ and moreover $|h| = q$ (leading to phase 2).

**Phase 2.** In the second phase of the evolution (Lemma 11), as long as $h \in \mathcal{H}_{<1/2}$ and $|h| = q$, due to Lemma 3, any shorter hypothesis will have noticeably smaller performance by the selection of tolerance. Thus, the only beneficial mutations are those that increase the number of good variables while retaining the size to be $q$ for the hypothesis. The smallest increase in performance on such beneficial swaps is obtained when only one bad variable is replaced by a good one in one step and the increase is $4p^{|h|+u-1}(1-p) = 2^{2-|h|-u} \geq 2^{1-2q}$. (Note also that if many good variables are brought into the hypothesis in one generation, such that the resulting hypothesis belongs to $\mathcal{H}_{1/2} \cup \mathcal{H}_{>1/2}$, then by Lemma 4 the increase in performance is at least $2^{-q}$ and by Lemma 6 the increase in performance is at least $3 \cdot 2^{-q}$. Both such increases are at least as large as $2^{1-2q}$ for any $q \geq 1$.) Hence we reach phase 3 ($h \in \mathcal{H}_{1/2}$) or phase 4 ($h \in \mathcal{H}_{>1/2}$) below.

**Phase 3.** In the third phase of the evolution (Lemma 12), a hypothesis $h_0$ has been formed that is missing precisely one variable from the target $c$. Due to Lemma 4, any other

short hypothesis $h \in \mathcal{H}_{<1/2}$ has noticeably smaller performance by an amount of at least $2^{-q}$.

In the case where the target is medium (that is, $|c| = q+1$), then $h_0$ is already a best $q$-approximation of $c$ since $h_0 \in \mathcal{H}_{1/2} \Rightarrow u = 1$. Due to our remark above, this formation is stable, as any hypothesis $h$ with $|h| < q$ will have $u \geq 2$, and moreover, among the hypotheses of size $q$ again there can not be more than 2 good variables missing from $h$ as this would imply $h \in \mathcal{H}_{<1/2}$ again.

In the case where the target is short, we have that $m = |c| - 1 \leq q - 1$. Therefore, $|h_0| \in \{|c|-1, |c|, \ldots, q\}$. As explained in the proof of Lemma 12 we are interested in introducing the last missing good variable to $h$, which can either happen by appending it to $h$ when $|h| < q$, or by performing a beneficial swap when there are redundant bad variables present in $h$. Either of these two mutations results in a new hypothesis $h' \in \mathcal{H}_{>1/2}$ and due to Lemma 5 the increase in performance is at least $2^{1-q}$. Similarly, until such a good event occurs, by the selection of tolerance, any hypothesis $h'' \in \mathcal{H}_{<1/2}$ has noticeably smaller performance compared to $h$ by an amount of at least $2^{-q}$ due to Lemma 4. Thus, when the target is short, with the application of Lemma 12, we reach phase 4.

**Phase 4.** In the fourth phase of the evolution (Lemma 13), a specialization of a short target has been formed. Removing or replacing one or more good variables from the hypothesis within one mutation, results in transitioning from a hypothesis $h \in \mathcal{H}_{>1/2}$ to a hypothesis $h' \in \mathcal{H}_{1/2} \cup \mathcal{H}_{<1/2}$ and due to Lemmas 5 and 6, the decrease in performance is at least $2^{1-q}$. By the selection of tolerance such mutations will be characterized as deleterious. Thus, the only beneficial mutations that can occur are those that delete one or more bad variables from $h$ in one step (without affecting the good variables). Deleting one such bad variable results in increase in performance by an amount of $2p^{|h|-1}(1-p) = 2^{1-|h|} \geq 2^{1-q}$.

Once we reach the target, any mutation that removes one or more good variables, due to Lemmas 5 and 6, will have as a result a noticeable decrease in performance by an amount of at least $2^{1-q}$. Similarly introducing one or more bad variables (which thus maintains that $h \in \mathcal{H}_{>1/2}$) also results in a decrease in performance by an amount $|\Delta| = 2p^{|h|}(1-p) = 2^{-|h|} \geq 2^{-q}$.

Thus, from all four phases, when a beneficial or deleterious mutation occurs, then the performance of the hypothesis is affected by an additive amount at least $2^{1-2q}$. Therefore, we set the tolerance $t$ to be

$$t = 2^{-2q}\,.$$

Due to Lemmas 10, 11, 12 and 13, evolution lasts not more than $\lceil 16enq/\delta \rceil + \lceil 16en^2q/\delta \rceil + \lceil 16en^2/\delta \rceil + \lceil 16enq/\delta \rceil \leq 4 + 32enq/\delta + 17en^2q/\delta \leq 53en^2q/\delta$ generations (regardless of the target) with failure probability, by a union bound, not more than $\delta/4$. The neighborhood in each generation has size not larger than 2. Hence, the total number of hypotheses that need to be estimated is not more than $106en^2q/\delta$. By the analysis above we want to estimate the performance of each hypothesis within $\epsilon_s = t/2$ of its true value.

Requiring $R \geq \left\lceil \frac{8}{t^2} \cdot \ln\left(\frac{284en^2q}{\delta^2}\right)\right\rceil$ samples for estimating the empirical performance of each hypothesis, it follows by Hoeffding's bound (Proposition 1), using $\alpha = -1$ and $\beta = 1$, that the empirical performance of each hypothesis is estimated within $\epsilon_s = t/2$ of its exact value with probability at least $1 - \delta_s = 1 - \delta^2/(142en^2q)$. As the number of different hypotheses is not more than $106en^2q/\delta$, by the union bound, the performance of *every* hypothesis in this phase is computed within $\epsilon_s = t/2$ of its exact value except with probability at most $\sum_{i=1}^{106en^2q/\delta} \delta^2/(142en^2q) \leq \sum_{i=1}^{106en^2q/\delta} 3\delta^2/(4 \cdot 106en^2q) = 3\delta/4$.

Thus, with a union bound argument again, the performance of each hypothesis is computed within $\epsilon_s = t/2$ of its true value and evolution achieves its goal within $\mathcal{O}\left(n^2q/\delta\right)$ generations with total probability at least $1 - \delta$.

Since $q = \lceil \log_2(3/\varepsilon) \rceil$ we have $2^{-q} \geq \frac{\varepsilon}{6}$ and hence $t = 2^{-2q} \geq \varepsilon^2/36$. The total sample size follows. □

#### 4.4.2 Complexity Analysis when $p \in (0, 1/3)$

**Theorem 6** (Evolution in $\mathcal{C}_n^{\leq q}$ when $p \in (0, 1/3)$)**.** *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ product distribution with parameter $p \in \mathbb{R}_{alg}$ such that $p \in (0, 1/3)$. Using the hypothesis class $\mathcal{H} = \mathcal{C}_n^{\leq q}$, starting from any initial hypothesis, the (1+1) EA will evolve a hypothesis $h$ such that $\mathbf{Pr}\left(Perf_{\mathcal{B}_n}(h, c) > 1 - \varepsilon\right) \geq 1 - \delta$ in $\mathcal{O}\left(n^2q/\delta\right)$ generations with total sample size $\widetilde{\mathcal{O}}\left(\frac{n^2q}{\delta \cdot \varepsilon^2 \cdot (\min\{4p\varepsilon/9, 1-3p\})^2}\right)$.*

*Proof.* Note that long targets are taken care of by Lemma 10 as soon as a hypothesis of size $q$ has been formed. Thus, below we will only discuss about short targets.

**Phase 1.** In the first phase of the evolution (Lemma 10), as long as $h \in \mathcal{H}_{<1/2}$, increasing the size of $h$ results in increase in performance by an amount of at least $2p^{q-1}(1-3p)$ due to Lemma 7. Lemma 10 serves its purpose when either $h \in \mathcal{H}_{>1/2}$ (leading to phase 3 below), or it is still the case that $h \in \mathcal{H}_{<1/2}$ and moreover $|h| = q$ (leading to phase 2). Of course it can happen that all the good variables are brought into the hypothesis in one generation; then by Lemma 9 the increase in performance is at least $\frac{8}{3}p^q$.

**Phase 2.** In the second phase of the evolution (Lemma 11), as long as $h \in \mathcal{H}_{<1/2}$ and $|h| = q$, due to Lemma 7, any shorter hypothesis will have noticeably smaller performance by the selection of tolerance. Thus, the only beneficial mutations are those that increase the number of good variables while retaining the size of the hypothesis to be $q$. The smallest increase in performance on such beneficial swaps is obtained when only one bad variable is replaced by a good one in one step and the increase is $4p^{|h|+u-1}(1-p) \geq 4p^{2q-1}(1-p) \geq 8p^{2q-1}/3$. Hence, we reach phase 3 once sufficiently many beneficial swaps have occurred.

**Phase 3.** In the third phase of the evolution (Lemma 13), a specialization of the target has been formed. Removing or replacing one or more good variables from the hypothesis results in transitioning from a hypothesis $h \in \mathcal{H}_{>1/2}$ to a hypothesis $h' \in \mathcal{H}_{<1/2}$; due to Lemma 9 the decrease in performance will be at least $\frac{8}{3}p^q$ and by the selection of the tolerance it will be characterized as deleterious. Thus, the only beneficial

mutations that can occur are those that delete one or more bad variables from $h$ in one step (without affecting the good ones). Deleting such a bad variable increases the performance by an amount of $2p^{|h|-1}(1-p) \geq 2p^{q-1}(1-p)$. In turn, for any $p \in (0, 1/3]$ we have that $2p^{q-1}(1-p) \geq 4p^{2q-1}(1-p)$.

Once we reach the target, any mutation that removes one or more good variables, due to Lemma 9 will have as a result a noticeable decrease in performance. Similarly introducing one or more bad variables (which thus maintains that $h \in \mathcal{H}_{>1/2}$) also results in a decrease in performance by an amount $|\Delta| = 2p^{|h|}(1-p) \geq 2p^{q-1}(1-p)$.

Thus, from all three phases, for any $0 < p \leq 1/3$, using the fact that $8p^{2q-1}/3$ is a lower bound for the three quantities $2p^{q-1}(1-p)$, $4p^{2q-1}(1-p)$, and $\frac{8}{3}p^q$, when a mutation occurs that affects the true performance, then the performance of a hypothesis changes by at least an additive factor of $|\Delta| = 2p^{q-1} \cdot \min\{4p^q/3,\ 1-3p\}$. We thus set the tolerance to be

$$t = p^{q-1} \cdot \min\{4p^q/3,\ 1-3p\}\ .$$

Due to Lemmas 10, 11 and 13, evolution lasts not more than $\lceil 16enq/\delta \rceil + \lceil 16en^2q/\delta \rceil + \lceil 16enq/\delta \rceil \leq 3 + 32enq/\delta + 16en^2q/\delta \leq 35enq/\delta + 16en^2q/\delta \leq 51en^2q/\delta$ generations (regardless of the target) with failure probability, by a union bound, not more than $3\delta/16$. The neighborhood in each generation has size not larger than 2. Hence, the total number of hypotheses that need to be estimated is not more than $102en^2q/\delta$. Furthermore, we want to estimate the performance of each hypothesis within $\epsilon_s = t/2$ of its true value.

Requiring $R \geq \left\lceil \frac{8}{t^2} \cdot \ln\left(\frac{252en^2q}{\delta^2}\right) \right\rceil$ samples for estimating the empirical performance of each hypothesis, it follows by Hoeffding's bound (Proposition 1), using $\alpha = -1$ and $\beta = 1$, that the empirical performance of each hypothesis is estimated within $\epsilon_s = t/2$ of its exact value except with probability at most $\delta_s = \delta^2/(126en^2q)$. As the number of different hypotheses is not more than $102en^2q/\delta$, by the union bound, the performance of *every* hypothesis in this phase is computed within $\epsilon_s = t/2$ of its exact value except with probability at most $\sum_{i=1}^{102en^2q/\delta} \delta^2/(126en^2q) \leq \sum_{i=1}^{102en^2q/\delta} 13\delta^2/(16 \cdot 102en^2q) = 13\delta/16$.

Thus, with a union bound argument again, the performance of each hypothesis is computed within $\epsilon_s = t/2$ of its true value and evolution achieves its goal within $\mathcal{O}\left(n^2q/\delta\right)$ generations except with probability at most $\delta$.

Since $q = \lceil \log_{1/p}(3/\varepsilon) \rceil$ we have $p^q \geq p^{1+\log_{1/p}(3/\varepsilon)} = p\varepsilon/3$ and hence $t = p^{q-1} \cdot \min\{4p^q/3, 1-3p\} \geq \frac{\varepsilon}{3} \cdot \min\{4p\varepsilon/9, 1-3p\}$. The sample size follows. □

#### 4.4.3 Complexity Analysis when $p = 1/3$

**Theorem 7** (Evolution in $\mathcal{C}_n^{\leq q}$ when $p = 1/3$)**.** *Let $\mathcal{B}_n$ be a Bernoulli$(p)^n$ product distribution with parameter $p = 1/3$. Using the hypothesis class $\mathcal{H} = \mathcal{C}_n^{\leq q}$, starting from any initial hypothesis, the (1+1) EA will evolve a hypothesis $h$ such that $\mathbf{Pr}\left(Perf_{\mathcal{B}_n}(h, c) > 1-\varepsilon\right) \geq 1-\delta$ in $\mathcal{O}\left(n^2q/\delta\right)$ generations with total sample size $\widetilde{\mathcal{O}}\left(\frac{n^2q}{\delta \cdot \varepsilon^4}\right)$.*

*Proof.* The only difference in the proof of this theorem compared to Theorem 6 is that for phases 1 and 2, instead of using Lemma 7 we use Lemma 8. Thus, the performance of a

hypothesis due to beneficial or deleterious mutations, is affected by at least an amount of $\min\{8 \cdot 3^{1-2q}/3, 4 \cdot 3^{-1-2q}\} = \frac{4}{3} \cdot 3^{-2q}$. Therefore, we set the tolerance $t$ to be

$$t = 2 \cdot 3^{-1-2q}\,.$$

Since $q = \lceil \log_3(3/\varepsilon) \rceil$, we have that $3^{-q} > 3^{-2} \cdot \varepsilon$. As a consequence from the above, for the tolerance we have $t = 2 \cdot 3^{-1-2q} \geq 2 \cdot 3^{-5} \cdot \varepsilon^2$. The sample size follows. □

## 5. Uniform Setup

Algorithm 3 provides a uniform setup so that the (1+1) EA can converge to an $\varepsilon$-optimal hypothesis *regardless of the value of $p$ that characterizes the underlying distribution, as long as $p$ belongs to a specific family.* The idea is that, on one hand Lemmas 1 and 2 hold for any sufficiently large frontier $q$ and on the other hand we will set the tolerance $t$ so that it lower bounds any tolerance that is used by Algorithm 2 when the exact value of $p$ is known and $p$ belongs to the particular family. Therefore, we will use the fact that Theorem 4 indicates that when $p = \frac{1}{2}$, then the tolerance $t$ should be $2^{-2q}$, as well as the fact that it should hold $t = p^{q-1} \cdot \min\{4p^q/3, 1-3p\}$ when $p \in (0, \frac{1}{3})$, and $t = 2 \cdot 3^{-1-2q}$ when $p = \frac{1}{3}$. As an additional consequence of the fact that Algorithm 3 is now agnostic to the underlying $p$ that characterizes the underlying distribution, this implies convergence for values of $p$ where $p$ is *transcendental* since *the actual value of $p$ is no longer part of the input.*

**Algorithm 3:** Mutator so that the (1+1) EA converges with a uniform setup for a class of $\mathcal{B}_{n,p}$ distributions.

**Input:** $n$, $\alpha \in (0, 3/13]$, $\delta \in (0,1)$, $\varepsilon \in (0,2)$, $h \in \mathcal{H}$
**Output:** a new hypothesis

$q \leftarrow \lceil \log_2(3/\varepsilon) \rceil$ ; /* set the frontier */
$k \leftarrow \lceil \log_2(1/\alpha) \rceil$ ; /* set the constant related to $\alpha$ */
$h' \leftarrow$ `Mutate(`$h$`)`; ; /* perform the mutation */
**if** $|h'| \leq q$ **then** $N \leftarrow \{h'\}$ ; /* set the neighborhood */
**else return** $h$;
$t \leftarrow \frac{52}{9} \cdot (\varepsilon/6)^{2k}$ ; /* set the tolerance */
$\delta_s \leftarrow \delta^2/(142en^2q)$ ; /* set the confidence for estimating the performance */
$\epsilon_s \leftarrow t/2$; ; /* set the approximation-error bound for estimating the performance */
$\nu_h \leftarrow$ `Perf(`$p$, $h$, $\epsilon_s$, $\delta_s$`)` ; /* estimate empirically the performance */
$\nu_{h'} \leftarrow$ `Perf(`$p$, $h'$, $\epsilon_s$, $\delta_s$`)` ; /* estimate empirically the performance */
/* apply the selection mechanism below and return the appropriate hypothesis */
**if** $\nu_{h'} > \nu_h + t$ **then return** $h'$;
**else if** $\nu_{h'} \geq \nu_h - t$ **then return** `USelect(`$\{h\} \cup \{h'\}$`)`;
**else return** $h$;

**Theorem 8** (Evolution with a Uniform Setup)**.** *Let $\alpha \in \mathbb{R}_{alg}$ with $0 < \alpha \leq 3/13$. Let $k = \lceil \log_2(1/\alpha) \rceil$. Let $q = \lceil \log_2(3/\varepsilon) \rceil$. Let $\mathcal{I} = [\alpha, 1/3 - 4\alpha/9] \cup \{1/3\} \cup \{1/2\}$. Let $p \in \mathcal{I}$. Consider a Bernoulli$(p)^n$ distribution $\mathcal{B}_n$ over $\{0,1\}^n$ that is characterized by $p$. Then, using the hypothesis class $\mathcal{H} = \mathcal{C}_n^{\leq q}$, starting from any initial hypothesis, the (1+1) EA will evolve a hypothesis $h$ such that $\mathbf{Pr}\left(Perf_{\mathcal{B}_n}(h,c) > 1-\varepsilon\right) \geq 1-\delta$ in $\mathcal{O}\left(n^2 q/\delta\right)$ generations with total sample size $\widetilde{\mathcal{O}}\left(6^{4k} n^2 q/(\delta \varepsilon^{4k})\right)$.*

*Proof.* Note that for any $0 < \varepsilon < 2$, we have $q \geq 1$.

**Case** $p \in \mathcal{I}_1 = [\alpha, 1/3 - 4\alpha/9]$**.** We first use the fact that $4\alpha/3$ is a lower bound for the quantity $1-3p$ for any $p \in \mathcal{J} = [3/13, 1/3 - 4\alpha/9]$. Thus, for any $p \in \mathcal{I}_1$ we have $\min\{4p^q/3, 1-3p\} \geq \min\{4p^q/3, 4\alpha/3\} \geq \min\{4\alpha^q/3, 4\alpha/3\} = 4\alpha^q/3$. It follows that for any $p \in \mathcal{I}_1$ we have $p^{q-1} \cdot \min\{4p^q/3, 1-3p\} \geq \alpha^{q-1} \cdot (4\alpha^q/3) \geq 52\alpha^{2q}/9$, where in the last inequality we used the fact that $\alpha \leq 3/13$. Furthermore, $2^{-k} \leq \alpha \Rightarrow \alpha^q \geq 2^{-kq} = 2^{-k\lceil \log_2(3/\varepsilon)\rceil} \geq \left(2^{-1-\log_2(3/\varepsilon)}\right)^k = (\varepsilon/6)^k$. Therefore, it suffices to set the tolerance $t$ so that,

$$t \leq \frac{52}{9} \cdot (\varepsilon/6)^{2k} \leq \frac{52}{9} \cdot \alpha^{2q}\,. \tag{16}$$

**Case** $p = 1/3$**.** We observe that $\frac{2}{3} \cdot 3^{-2q} \geq \frac{2}{3} \cdot \left(3^{-1-\log_2(3/\varepsilon)}\right) = \frac{2}{27} \cdot \left(3^{-\log_3(3/\varepsilon)}\right)^{2/\log_3(2)} = \frac{2}{27} \cdot (\varepsilon/3)^{2/\log_3(2)} \geq \frac{2}{27} \cdot (\varepsilon/3)^{3.17}$. Hence, it suffices to set,

$$t \leq \frac{2}{27} \cdot (\varepsilon/3)^{3.17} \leq \frac{2}{3} \cdot 3^{-2q}\,. \tag{17}$$

**Case** $p = 1/2$**.** Since $2^{-2q} \geq (\varepsilon/6)^2$ it suffices to set,

$$t \leq (\varepsilon/6)^2 \leq 2^{-2q}\,. \tag{18}$$

Now, using the fact that $\varepsilon < 2$ and $k \geq 3$, the first observation is, $\frac{52}{9} \cdot (\varepsilon/6)^{2k} \leq \frac{52}{9} \cdot (\varepsilon/6)^6 = \frac{52}{9} \cdot (\varepsilon/6)^4 \cdot (\varepsilon/6)^2 \leq \frac{52}{9} \cdot (2/6)^4 \cdot (\varepsilon/6)^2 \leq \frac{52}{9^3} \cdot (\varepsilon/6)^2 < \varepsilon^2/36$. The second observation is, $\varepsilon^{2.83} < 2^{2.83} < \frac{2 \cdot 9 \cdot 6^6}{52 \cdot 27 \cdot 3^{3.17}}$ and thus $\frac{52}{9} \cdot (\varepsilon/6)^{2k} \leq \frac{52}{9} \cdot (\varepsilon/6)^6 = \frac{52}{9 \cdot 6^6} \cdot \varepsilon^{2.83} \cdot \varepsilon^{3.17} < \frac{2}{27} \cdot (\varepsilon/3)^{3.17}$. Hence, by (16), (17) and (18), it suffices to set, $t = \frac{52}{9} \cdot (\varepsilon/6)^{2k}$ unconditionally, so that evolution can achieve its goal with a uniform setup for every $p \in \mathcal{I}$. □

## 6. Conclusion

We examined a modification of the well-studied evolutionary mutation mechanism that is used in evolutionary algorithms, within the framework of evolvability. We modified the typical version of the (1+1) EA so that we can cope with noisy estimates of the fitness function. Our analysis was performed under a set of $Bernoulli(p)^n$ distributions.

Exploring such intuitive mutation mechanisms is desirable towards forming a better theory for evolving functions and in fact this is a property that is sought for explicitly (see, e.g., Lissovoi & Oliveto, 2019; Reyzin, 2020). The convergence relied on a new characterization of the fitness landscape of monotone conjunctions by providing different *fitness levels*. These fitness levels allow the analysis to be decomposed into different phases, as the evolved function migrates from levels that have lower fitness values to levels that have higher fitness values. In addition, we provided a distribution-specific result as well as a distribution-free result for a class of distributions.

A very natural open question is whether the evolutionary mutation mechanism allows convergence for a broader set of distributions as, for example, the swapping algorithm in the analysis by Diochnos (2016) does. Are there other intuitive evolutionary mechanisms that provably allow us to cover a broader set of distributions for monotone conjunctions? Can similar intuitive evolutionary mechanisms provide results for other concept classes? Finally, as one of the reviewers suggested, perhaps an interesting direction is to explore the convergence in a population-based evolutionary algorithm, potentially allowing self-adjusting the mutation rate along the lines of the work of Doerr, Gießen, Witt and Yang (2019).

## Appendix A. Definition of Evolvability

For the performance and the empirical performance we use (1) and (2) respectively. Below we draw the definitions from the work of Valiant (2009); however, we include the failure probability $\delta$ explicitly.

**Definition 2** (Neighborhood and Selection). *For a polynomial $p(\cdot,\cdot,\cdot)$ and a representation class $R_n$ a $p$-neighborhood $N$ on $R_n$ is a pair $M_1, M_2$ of randomized polynomial time Turing machines such that the numbers $n$ (in unary), $\lceil 1/\varepsilon \rceil$, $\lceil 1/\delta \rceil$ and a representation $r \in R_n$ act as follows: $M_1$ outputs all the members of a set $Neigh_N(r,\varepsilon,\delta) \subseteq R_n$, that contains $r$ and may depend on random coin tosses of $M_1$, and has size at most $p(n, 1/\varepsilon, 1/\delta)$. If $M_2$ is then run on this output of $M_1$, it in turn outputs one member of $Neigh_N(r,\varepsilon,\delta)$, with member $r_1$ being output with a probability $\mathbf{Pr}_N(r, r_1) \geq 1/p(n, 1/\varepsilon, 1/\delta)$.*

**Definition 3** (Selection Mechanism Details). *For confidence parameter $\delta$, error parameter $\varepsilon$, positive integers $n$ and $s$, an ideal function $f \in \mathcal{C}_n$, a representation class $R_n$ with $p(n, 1/\varepsilon, 1/\delta)$-neighborhood $N$ on $R_n$, a distribution $D_n$, a representation $r \in R_n$ and a real number $t$, the mutator $Mu(f, p(n, 1/\varepsilon, 1/\delta), R_n, N, D_n, s, r, t)$ is a random variable that on input $r \in R_n$ takes a value $r_1 \in R_n$ determined as follows: For each $r_1 \in Neigh_N(r,\varepsilon,\delta)$ it first computes an empirical value of $\nu(r_1) = Perf_{D_n}(r_1, f, s)$. Let Bene be the set $\{r_1 \mid \nu(r_1) > \nu(r) + t\}$ and Neut be the set difference $\{r_1 \mid \nu(r_1) \geq \nu(r) - t\} \setminus Bene$. If $Bene \neq \emptyset$ then output $r_1 \in Bene$ with probability $\mathbf{Pr}_N(r, r_1) / \sum_{r_1 \in Bene} \mathbf{Pr}_N(r, r_1)$. Otherwise ($Bene = \emptyset$), output an $r_1 \in Neut$, the probability of a specific $r_1$ being*

$$\mathbf{Pr}_N(r, r_1) / \sum_{r_1 \in Neut} \mathbf{Pr}_N(r, r_1) \ .$$

**Definition 4** (One Evolutionary Step). *For a mutator $Mu(f, p(n, 1/\varepsilon, 1/\delta), R_n, N, D_n, s, r, t)$ a $t$-evolution step on input $r_1 \in R_n$ is the random variable $r_2 = Mu(f, p(n, 1/\varepsilon, 1/\delta), R_n, N, D_n, s, r_1, t)$. We then say $r_1 \to r_2$ or $r_2 \leftarrow Evolve(f, p(n, 1/\varepsilon, 1/\delta), R_n, N, D_n, s, r_1, t)$.*

We say that polynomials $t_\ell(x, y, z)$ and $t_u(x, y, z)$ are *polynomially related* if for some $\eta > 1$ for all $x, y, z$ such that $(0 < x, y, z < 1)$ it holds $(t_u(x, y, z))^\eta \leq t_\ell(x, y, z) \leq t_u(x, y, z)$.

**Definition 5** (An Evolutionary Sequence (where for Each Step $i$ it holds $t_i \in [t_\ell, t_u]$)). *For a mutator $Mu(f, p(n, 1/\varepsilon, 1/\delta), R_n, N, D_n, s, r, t)$ a $(t_\ell, t_u)$-evolution sequence for $r_1 \in R_n$ is a random variable that takes as values sequences $r_1, r_2, r_3, \ldots$ such that for all $i$, $r_i \leftarrow Evolve(f, p(n, 1/\varepsilon, 1/\delta), R_n, N, D_n, s, r_{i-1}, t_i)$, where $t_\ell(1/n, \varepsilon, \delta) \leq t_i \leq t_u(1/n, \varepsilon, \delta)$, $t_\ell$ and $t_u$ are polynomially related polynomials, and $t_i$ is the output of a TM $T$ on input $r_{i-1}, n$, $\varepsilon$, $\delta$.*

**Definition 6** (Goal of Evolution; Evolvability of Concept Class $\mathcal{C}_n$ with an Evolutionary Sequence). *For polynomials $p(n, 1/\varepsilon, 1/\delta)$, $s(n, 1/\varepsilon, 1/\delta)$, $t_\ell(1/n, \varepsilon, \delta)$ and $t_u(1/n, \varepsilon, \delta)$, a representation class $R_n$ and $p(n, 1/\varepsilon, 1/\delta)$-neighborhood $N$ on $R_n$, the class $\mathcal{C}_n$ is $(t_\ell, t_u)$-evolvable by $(p(n, 1/\varepsilon, 1/\delta), R_n, N, s(n, 1/\varepsilon, 1/\delta))$ over distribution $D_n$ if there is a polynomial $g(n, 1/\varepsilon, 1/\delta)$ and a Turing machine $T$, which computes a tolerance bounded between $t_\ell$ and $t_u$, such that for every positive integer $n$, every $f \in \mathcal{C}_n$, every $\delta > 0$, every $\varepsilon > 0$, and every $r_0 \in R_n$ it is the case that with probability at least $1 - \delta$, a $(t_\ell, t_u)$-evolution sequence $r_0, r_1, r_2, \ldots$, where $r_i \leftarrow Evolve(f, p(n, 1/\varepsilon, 1/\delta), R_n, N, D_n, s(n, 1/\varepsilon, 1/\delta), r_{i-1}, T(r_{i-1}, n, \varepsilon))$, will have $Perf_{D_n}\left(r_{g(n,1/\varepsilon,1/\delta)}, f\right) \geq 1 - \varepsilon$.*

**Definition 7.** *A class $\mathcal{C}_n$ is evolvable by $(p(n, 1/\varepsilon, 1/\delta), R_n, N, s(n, 1/\varepsilon, 1/\delta))$ over $D_n$ iff for some pair of polynomially related polynomials $t_\ell, t_u$, $\mathcal{C}_n$ is $(t_\ell, t_u)$-evolvable by $(p(n, 1/\varepsilon,$-$1/\delta), R_n, N, s(n, 1/\varepsilon, 1/\delta))$ over $D_n$.*

**Definition 8.** *A class $\mathcal{C}_n$ is evolvable by $R_n$ over $D_n$ iff for some polynomials $(p(n, 1/\varepsilon, 1/\delta)$ and $s(n, 1/\varepsilon, 1/\delta))$, and some $p\,(n, 1/\varepsilon, 1/\delta)$-neighborhood $N$ on $R_n$, $\mathcal{C}_n$ is evolvable by $(p(n, 1/\varepsilon, 1/\delta), R_n, N, s(n, 1/\varepsilon, 1/\delta))$ over $D_n$.*

## Appendix B. Bounded and Unbounded Models of Evolution

We make the following remarks on different models of evolution.

**Definition 9** (Bounded-/Unbounded- Precision Evolution)**.** *The unbounded-precision model occurs for $t_\ell = 0$ in the definitions of evolvability. The bounded-precision model occurs for $t_\ell > 0$.*

The bounded precision model allows intermediate setups between black-box optimization (unbounded-precision) and evolvability, where in evolvability one can determine the sign of the performance difference $\Delta = \mathrm{Perf}_{D_n}(h', c) - \mathrm{Perf}_{D_n}(h, c)$, if the two fitness values differ significantly; that is, $\Delta$ is $poly(1/n, \varepsilon, \delta)$. Bounded-precision oracles are of interest in other domains as well (see, e.g., Ajtai, Feldman, Hassidim, & Nelson, 2016). Furthermore, in the bounded-precision model it might be the case that the tolerances $t_\ell$ and $t_u$ are not polynomially related. For example, theoretically it is possible to allow $t_u$ to have some value larger than 1, since the correlation between the hypothesis $h$ and the target $c$ is between -1 and +1.

## References


Ajtai, M., Feldman, V., Hassidim, A., & Nelson, J. (2016). Sorting and selection with imprecise comparisons. *ACM Transactions on Algorithms*, *12*(2), 19.

Angelino, E., & Kanade, V. (2014). Attribute-efficient evolvability of linear functions. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pp. 287–300.

Astete-Morales, S., Cauwet, M.-L., & Teytaud, O. (2015). Evolution Strategies with Additive Noise: A Convergence Rate Lower Bound. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, Aberystwyth, United Kingdom, January 17 - 20, 2015*, pp. 76–84.

Ben-David, S., & Dichterman, E. (1998). Learning with restricted focus of attention. *Journal of Computer and System Sciences*, *56*(3), 277–298.

Ben-David, S., Itai, A., & Kushilevitz, E. (1995). Learning by Distances. *Information and Computation*, *117*(2), 240–250.

Benedek, G. M., & Itai, A. (1991). Learnability with Respect to Fixed Distributions. *Theoretical Computer Science*, *86*(2), 377–390.

Bshouty, N. H., & Feldman, V. (2002). On Using Extended Statistical Queries to Avoid Membership Queries. *Journal of Machine Learning Research*, *2*, 359–395.

Cai, W., & Shao, X. (2002). A fast annealing evolutionary algorithm for global optimization. *Journal of Computational Chemistry*, *23*(4), 427–435.

Cordón, O., de Moya Anegón, F., & Zarco, C. (2002). A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems. *Soft Computing*, *6*(5), 308–319.

Corus, D., Dang, D., Eremeev, A. V., & Lehre, P. K. (2018). Level-Based Analysis of Genetic Algorithms and Other Search Processes. *IEEE Transactions on Evolutionary Computation*, *22*(5), 707–719.

Dang, D.-C., & Lehre, P. K. (2015). Efficient Optimisation of Noisy Fitness Functions with Population-based Evolutionary Algorithms. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, Aberystwyth, United Kingdom, January 17 - 20, 2015*, pp. 62–68.

Diochnos, D. I. (2016). On the Evolution of Monotone Conjunctions: Drilling for Best Approximations. In *Algorithmic Learning Theory - 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, pp. 98–112.

Diochnos, D. I., & Turán, G. (2009). On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance. In *Stochastic Algorithms: Foundations and Applications, 5th International Symposium, SAGA 2009, Sapporo, Japan, October 26-28, 2009. Proceedings*, pp. 74–88.

Doerr, B., Gießen, C., Witt, C., & Yang, J. (2019). The $(1 + \lambda)$ Evolutionary Algorithm with Self-Adjusting Mutation Rate. *Algorithmica*, *81*(2), 593–631.

Domingos, P. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, New York.

Droste, S. (2004). Analysis of the (1+1) EA for a Noisy OneMax. In *Genetic and Evolutionary Computation - GECCO 2004, Genetic and Evolutionary Computation Conference, Seattle, WA, USA, June 26-30, 2004, Proceedings, Part I*, pp. 1088–1099.

Droste, S., Jansen, T., & Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, *276*(1-2), 51–81.

Duda, R. O., & Shortliffe, E. H. (1983). Expert Systems Research. *Science*, *220*, 261–268.

Feldman, V. (2008). Evolvability from learning algorithms. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pp. 619–628.

Feldman, V. (2009). Robustness of Evolvability. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, pp. 277–292.

Feldman, V. (2011). Distribution-Independent Evolvability of Linear Threshold Functions. In *COLT 2011 - The 24th Annual Conference on Learning Theory, June 9-11, 2011, Budapest, Hungary*, pp. 253–272.

Feldman, V. (2012). A Complete Characterization of Statistical Query Learning with Applications to Evolvability. *Journal of Computer and System Sciences*, *78*(5), 1444–1459.

Friedrich, T., & Neumann, F. (2017). What's Hot in Evolutionary Computation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 5064–5066.

Gießen, C., & Kötzing, T. (2016). Robustness of Populations in Stochastic Environments. *Algorithmica*, *75*(3), 462–489.

Gutjahr, W. J., & Pflug, G. C. (1996). Simulated Annealing for noisy cost functions. *Journal of Global Optimization*, *8*(1), 1–13.

Hoeffding, W. (1963). Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, *58*(301), 13–30.

Holland, J. H. (1986). Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. In Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.), *Machine Learning, An Artificial Intelligence Approach (Volume II)*, chap. 20, pp. 593–623. Morgan Kaufmann, Los Alamos, CA.

Kalai, A. T., & Vempala, S. (2006). Simulated annealing for convex optimization. *Mathematics of Operations Research*, *31*(2), 253–266.

Kanade, V. (2011). Evolution with Recombination. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pp. 837–846.

Kanade, V., Valiant, L. G., & Vaughan, J. W. (2010). Evolution with Drifting Targets. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pp. 155–167.

Kearns, M. J. (1998). Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, *45*(6), 983–1006.

Kötzing, T., Neumann, F., & Spöhel, R. (2011). PAC Learning and Genetic Programming. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*, pp. 2091–2096.

Koza, J. R. (1993). *Genetic programming - on the programming of computers by means of natural selection*. Complex adaptive systems. MIT Press.

Laird, P. D. (1988). *Learning from Good and Bad Data*. Kluwer Academic Publishers, Boston.

Lissovoi, A., & Oliveto, P. S. (2019). On the Time and Space Complexity of Genetic Programming for Evolving Boolean Conjunctions. *Journal of Artificial Intelligence Research*, *66*, 655–689.

Michael, L. (2012). Evolvability via the Fourier transform. *Theoretical Computer Science*, *462*, 88–98.

Mitchell, T. M. (1997). *Machine learning*. McGraw Hill series in computer science. McGraw-Hill.

Mühlenbein, H., & Mahnig, T. (2001). Evolutionary algorithms: From recombination to search distributions. In Kallel, L., Naudts, B., & Rogers, A. (Eds.), *Theoretical Aspects of Evolutionary Computing*, pp. 135–173. Springer Berlin Heidelberg, Berlin, Heidelberg.

Prugel-Bennett, A., Rowe, J., & Shapiro, J. (2015). Run-Time Analysis of Population-Based Evolutionary Algorithm in Noisy Environments. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, Aberystwyth, United Kingdom, January 17 - 20, 2015*, pp. 69–75.

Qian, C., Bian, C., Jiang, W., & Tang, K. (2019). Running Time Analysis of the (1+1)-EA for OneMax and LeadingOnes Under Bit-Wise Noise. *Algorithmica*, *81*(2), 749–795.

Quinlan, J. R. (1986). The Effect of Noise on Concept Learning. In Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.), *Machine Learning, An Artificial Intelligence Approach (Volume II)*, chap. 6, pp. 149–166. Morgan Kaufmann, Los Alamos, CA.

Reyzin, L. (2020). Statistical Queries and Statistical Algorithms: Foundations and Applications. *CoRR*, *abs/2004.00557*.

Ros, J. P. (1992). Learning Boolean Functions with Genetic Algorithms: A PAC Analysis. In *Proceedings of the Second Workshop on Foundations of Genetic Algorithms. Vail, Colorado, USA, July 26-29 1992*, pp. 257–275.

Sloan, R. H. (1995). Four Types of Noise in Data for PAC Learning. *Information Processing Letters*, *54*(3), 157–162.

Snir, S., & Yohay, B. (2019a). Extending the Evolvability Model to the Prokaryotic World: Simulations and Results on Real Data. *Journal of Computational Biology*, *26*(8), 794–805.

Snir, S., & Yohay, B. (2019b). Prokaryotic evolutionary mechanisms accelerate learning. *Discrete Applied Mathematics*, *258*, 222–234.

Sudholt, D. (2010). General Lower Bounds for the Running Time of Evolutionary Algorithms. In *Parallel Problem Solving from Nature - PPSN XI, 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*, pp. 124–133.

Valiant, L. (2013). *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books, Inc., New York, NY, USA.

Valiant, L. G. (1984). A Theory of the Learnable. *Communications of the ACM*, *27*(11), 1134–1142.

Valiant, L. G. (2009). Evolvability. *Journal of the ACM*, *56*(1), 3:1–3:21.

Valiant, P. (2014). Evolvability of Real Functions. *ACM Transactions on Computation Theory*, *6*(3), 12:1–12:19.

Watson, R. A., & Szathmáry, E. (2016). How can evolution learn?. *Trends in Ecology & Evolution*, *31*(2), 147–157.

Wegener, I., & Witt, C. (2005). On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions. *Journal of Discrete Algorithms*, *3*(1), 61–78.

Yap, C. (2000). *Fundamental problems of algorithmic algebra*. Oxford University Press.