ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Linear Evaluation Complexity of Surrogate-Assisted $(1+1)$-EA on OneMax

Oliver Kramer

Computational Intelligence Group
Department of Computer Science
University of Oldenburg, Germany
oliver.kramer@uni-oldenburg.de

Abstract. Fitness evaluations often dominate the runtime of evolutionary algorithms (EAs), yet most runtime analyses assume that every offspring is evaluated on the true objective function. This work presents a unified runtime-theoretic framework for *surrogate-assisted* evolutionary algorithms that selectively schedule true evaluations based on predictive information. The framework characterizes expected optimization time directly in terms of evaluation frequency and improvement probabilities, abstracting from specific surrogate implementations. Focusing on a surrogate model with imperfect prediction accuracy, we show that linear $\Theta(n)$ scaling of true fitness evaluations can be achieved on OneMax when accuracy remains bounded away from zero and evaluations are performed periodically with logarithmic frequency. Simulation results confirm this prediction and clearly separate the surrogate-assisted process from the classical $\Theta(n \log n)$ behavior of the $(1+1)$-EA.

1 Introduction

In evolutionary algorithms, fitness evaluations often dominate runtime, especially in settings where objective queries involve expensive simulations or physical experiments. To mitigate this cost, *surrogate-assisted optimization* uses predictive models to approximate the objective function and reduce the number of true evaluations [1, 2]. Models such as Gaussian processes, radial basis functions, and neural networks have been effectively integrated into evolutionary strategies and genetic algorithms for engineering and robotics tasks.

Despite this empirical success, theoretical understanding remains limited. Most runtime analyses still assume that every offspring is evaluated on the true objective. Classical results, such as sharp bounds for the $(1+1)$-EA on OneMax [3, 4], drift-based analyses of population schemes [5], and tight results for adaptive algorithms like the $(1+\lambda,\lambda)$-GA [6, 7], explain selection and mutation dynamics but not predictive or selective evaluation mechanisms.

Recent progress was made by Zhang et al. [8], who analyzed the $(1+1)$-EA with surrogate models on OneMax and LeadingOnes. Their drift-based proof shows that small surrogate errors can yield nearly linear optimization time, but their model assumes a fixed surrogate mechanism with predefined error characteristics.

This work develops a more general theoretical framework. Rather than analyzing a specific surrogate construction, we formalize surrogate mechanisms

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

as stochastic processes that govern both mutation and probabilistic evaluation. The framework expresses expected runtime in terms of evaluation frequency and improvement probability, providing a general formulation that also encompasses other surrogate-based mechanisms. Within this setting, we study a surrogate-assisted (1+1)-EA with imperfect but bounded prediction accuracy and show that linear scaling of true evaluations arises under broad, realistic conditions on OneMax. Experiments confirm these asymptotic predictions and illustrate how predictive knowledge can eliminate the logarithmic slowdown of mutation-only search.

2 Background

We consider the classical *OneMax* problem

$$f(x) = \sum_{i=1}^{n} x_i, \qquad x \in \{0,1\}^n,$$

which counts the number of one-bits in a binary string and is maximized by $x^* = (1, \ldots, 1)$. The (1+1)-EA starts from a random bit string and iteratively mutates it by flipping each bit with probability $1/n$. The offspring replaces the parent if its fitness is not worse. This simple scheme models the essence of stochastic hill-climbing and serves as the standard baseline for runtime analysis.

The optimization process is summarized in Algorithm 1. It operates on discrete fitness levels, gradually increasing the number of correct bits until the optimal string is reached.

Algorithm 1 Standard (1+1)-EA

```
Initialize x ∈ {0,1}^n uniformly at random
while f(x) < n do
    x' ← Mutate(x, 1/n)
    if f(x') ≥ f(x) then
        x ← x'
    end if
end while
```

Let $A_i = \{x \mid f(x) = i\}$ denote the set of all solutions with exactly i one-bits, referred to as *fitness level* i. During optimization, the algorithm moves sequentially from lower to higher fitness levels until it reaches the optimum A_n. During the optimization process, the algorithm moves stepwise from lower to higher levels until it reaches A_n, the optimal set. The *fitness-level method* [3] expresses the expected optimization time as

$$E[T] = \sum_{i=0}^{n-1} \frac{1}{p_i},$$

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

where p_i is the probability of leaving level A_i in a single iteration, i.e., the probability that mutation produces an offspring with higher fitness than the current solution. For the (1+1)-EA on OneMax, an improvement occurs when one of the $(n-i)$ incorrect bits flips and all other bits remain unchanged. The corresponding probability is

$$p_i = \frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1} \approx \frac{n-i}{en}.$$

The expected time to increase fitness from i to $i+1$ is therefore $1/p_i$, and summing over all i gives

$$E[T] \approx en \sum_{i=1}^{n} \frac{1}{i} = en \ln n + O(n),$$

which establishes the well-known $\Theta(n \log n)$ runtime of the (1+1)-EA on OneMax [4].

3 Surrogate-Efficiency Expression

A surrogate mechanism M is modeled as a stochastic process that modifies either the generation or the evaluation of offspring in an evolutionary algorithm. It encapsulates predictive information learned during the search and determines when true evaluations are required. At iteration t, given the current solution x_t and search history H_t, the surrogate defines a joint sampling rule

$$(\tilde{x}_{t+1}, \delta_{t+1}) \sim M(x_t, H_t),$$

where $\tilde{x}_{t+1}$ is the offspring and $\delta_{t+1} \in \{0,1\}$ indicates whether the offspring is evaluated on the true objective function. The parameters $p_i(M)$ and $\rho_i(M)$ denote, respectively, the probability of leaving fitness level i and the probability that an offspring generated on this level is actually evaluated.

Theorem 1 (Surrogate-Efficiency Expression)**.** *For any evolutionary algorithm with probabilistic evaluation or mutation control, the expected number of true fitness evaluations is*

$$E[T_M] = \sum_{i=0}^{n-1} \frac{\rho_i(M)}{p_i(M)}. \tag{1}$$

Proof. By the fitness-level method, the expected number of iterations spent on level A_i before an improvement occurs is $1/p_i(M)$. Since only a fraction $\rho_i(M)$ of offspring are evaluated on the true function, the expected number of real evaluations required to leave A_i is $\rho_i(M)/p_i(M)$. Summing over all fitness levels yields the total expected number of true evaluations. □

The expression (1) generalizes classical runtime analysis: the baseline (1+1)-EA corresponds to $\rho_i(M) = 1$ and $p_i(M) = p_i$, yielding $E[T] = en \ln n + O(n)$ on OneMax. Different surrogate mechanisms alter these probabilities to capture the effects of selective evaluation or predictive guidance.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

4 Surrogate Mechanism with Imperfect Accuracy

A surrogate mechanism replaces costly fitness evaluations with predicted values $\hat{f}(x)$ obtained from a learned approximation, such as a regression model, Gaussian process, or neural network. Instead of evaluating every offspring on the true objective, the algorithm alternates between phases of surrogate-based prediction and occasional true evaluations that recalibrate the surrogate model. This reflects the practical design of surrogate-assisted evolutionary algorithms used in expensive black-box optimization, where each true evaluation may correspond to a costly simulation or physical experiment.

The resulting optimization process is summarized in Algorithm 2, which extends the standard (1+1)-EA (Algorithm 1) by introducing a probabilistic surrogate accuracy parameter γ and an evaluation interval k.

Algorithm 2 Surrogate-Assisted (1+1)-EA with Imperfect Accuracy

Initialize $x \in \{0,1\}^n$, surrogate model $\hat{f}$
for each iteration $t = 1, 2, \ldots$ **do**
  if $t \bmod k = 0$ **then**
    Evaluate $f(x)$ and update $\hat{f}$
  end if
  Generate $x' \leftarrow \text{Mutate}(x, 1/n)$
  if rand() $< \gamma$ and $\hat{f}(x') \geq \hat{f}(x)$ **then**
    $x \leftarrow x'$ {Accepted by surrogate (correct prediction)}
  else if rand() $\geq \gamma$ and $f(x') \geq f(x)$ **then**
    $x \leftarrow x'$ {Accepted by true evaluation (correction step)}
  end if
end for

Here, $\gamma \in [0,1]$ denotes the probability that the surrogate correctly predicts the outcome of a comparison between two candidate solutions, i.e., the probability that it preserves the true fitness ranking. After each true evaluation, the surrogate is trusted for k predicted iterations on average, resulting in an effective evaluation probability $\rho_i(M_S) = 1/k$.

Because surrogate predictions may be imperfect, the effective probability of improvement is reduced to $p_i(M_S) = \gamma p_i$.

Substituting these quantities into Equation (1) yields the expected number of true evaluations,

$$E[T_{M_S}] = \sum_i \frac{1/k}{\gamma p_i} = \frac{1}{\gamma k} E[T] = O\left(\frac{n \log n}{\gamma k}\right),$$

showing that runtime decreases with both increasing surrogate accuracy γ and increasing surrogate usage k.

Theorem 2 (Linear Runtime for High-Accuracy Surrogates)**.** *If* $\gamma = \Theta(1)$ *and* $k = \Theta(\log n)$ *surrogate iterations are executed per true evaluation, the expected*

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

number of true evaluations satisfies

$$E[T_{M_S}] = O(n).$$

Proof. From the surrogate-efficiency expression $E[T_{M_S}] = (1/(\gamma k))E[T]$ and the classical result $E[T] = \Theta(n \log n)$, we obtain $E[T_{M_S}] = \Theta((n \log n)/(\gamma k))$. For surrogates of constant accuracy $\gamma = \Theta(1)$ and $k = \Theta(\log n)$ predicted steps per evaluation, the logarithmic term cancels, yielding linear scaling $E[T_{M_S}] = \Theta(n)$. □

This result formalizes the key trade-off in surrogate-assisted evolutionary optimization: frequent true evaluations maintain accuracy but increase cost, whereas longer surrogate phases risk incorrect guidance. When the surrogate achieves constant predictive reliability, it can safely replace a logarithmic fraction of true evaluations, reducing the expected number of real fitness queries to linear order without loss of convergence speed.

5 Experiments

The experiments empirically validate the theoretical predictions for the surrogate-assisted (1+1)-EA on the *OneMax* problem. We measure the number of true fitness evaluations, the key runtime metric in the analysis, and compare the classical (1+1)-EA to a surrogate-assisted variant that performs one true evaluation after every k surrogate-guided iterations. The surrogate is simulated with a fixed prediction accuracy γ, representing the probability that a proposed offspring is correctly identified as improving.

The block size is chosen as $k = \lceil \log n \rceil$, following the theoretical condition for linear expected evaluation complexity, and the surrogate accuracy is fixed to $\gamma = 0.9$ across all experiments. For each problem dimension $n \in \{50, 100, 200, 400, 800, 1600\}$, results are averaged over 200 independent runs.

Table 1 reports the mean number of true objective evaluations required to reach the optimum.

n	(1+1)-EA	Surrogate	k	n	(1+1)-EA	Surrogate	k
50	455.6	132.1	4	400	5763.4	1130.5	6
100	1064.1	248.4	5	800	12996.1	2238.6	7
200	2528.3	595.2	5	1600	28831.3	4970.6	7

Table 1: Mean number of true fitness evaluations for the classical (1+1)-EA and the surrogate-assisted variant at different problem sizes n.

The baseline algorithm exhibits the expected superlinear growth consistent with $\Theta(n \log n)$ behavior on OneMax. In contrast, the surrogate-assisted variant shows near-linear scaling in n. Normalizing the evaluation counts by n yields approximately constant values across problem sizes, confirming $\Theta(n)$ behavior in agreement with the theoretical analysis. Moreover, the surrogate consistently

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

reduces the number of true evaluations by more than an order of magnitude compared to the baseline across all tested dimensions.

6 Conclusions

This work establishes a quantitative connection between surrogate prediction accuracy, evaluation scheduling, and runtime efficiency in evolutionary algorithms. The central surrogate-efficiency expression (1) characterizes the expected number of true fitness evaluations in terms of improvement probabilities and evaluation frequencies, enabling a runtime analysis of predictive guidance that is independent of specific learning-model implementations.

For a surrogate-assisted (1+1)-EA with imperfect but bounded prediction accuracy, the theoretical analysis predicts linear $\Theta(n)$ scaling of true evaluations on the *OneMax* problem when evaluations are performed periodically with logarithmic frequency. Empirical results confirm this prediction, showing a clear deviation from the classical $\Theta(n \log n)$ behavior of the baseline (1+1)-EA across all tested problem sizes.

These findings demonstrate that predictive information, even when noisy, can fundamentally reduce evaluation complexity without assuming oracle-level accuracy. The proposed framework provides a principled foundation for the runtime analysis of surrogate-assisted evolutionary algorithms and identifies the conditions under which learned models can asymptotically accelerate evolutionary search.

References

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

[2] Dudy Lim, Yaochu Jin, Yew-Soon Ong, and Bernhard Sendhoff. Generalizing surrogate-assisted evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 14(3):329–355, 2010.

[3] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the $(1 + 1)$ evolutionary algorithm. *Theoretical Computer Science*, 276(1–2):51–81, 2002.

[4] Hsien-Kuei Hwang and Carsten Witt. Sharp bounds on the runtime of the $(1 + 1)$ ea via drift analysis and analytic combinatorial tools. In *Foundations of Genetic Algorithms (FOGA)*, pages 1–12. ACM, 2019.

[5] Pietro S. Oliveto and Carsten Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 545:2–19, 2014.

[6] Benjamin Doerr and Carola Doerr. A tight runtime analysis of the $(1 + (\lambda, \lambda))$ genetic algorithm on OneMax. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 1423–1430. ACM, 2015.

[7] Maxim Buzdalov and Benjamin Doerr. Runtime analysis of the $(1 + (\lambda, \lambda))$ genetic algorithm on random satisfiable 3-cnf formulas. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 1343–1350. ACM, 2017.

[8] Chao Zhang, Chao Bian, and Chao Qian. Runtime analysis of the (1+1) ea using surrogate models on OneMax and LeadingOnes. In *Parallel Problem Solving from Nature (PPSN)*, volume 13399 of *Lecture Notes in Computer Science*, pages 315–329. Springer, 2022.