Surrogate Model Assisted Evolutionary Algorithms: Performance Bound and Incremental Gaussian Process Model Updates

Hong Huang
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
hn582183@dal.ca

Dirk V. Arnold
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
dirk@dal.ca

ABSTRACT

The performance of surrogate model assisted algorithms for black-box optimization is impacted by two factors: algorithmic design choices on how to use the surrogate model, and the ability of the model to accurately represent the true objective function. In an effort to better understand the potential of surrogate model assisted algorithms, we propose to decouple those factors by studying the performance of the algorithms assuming perfect models. As a result, we obtain a natural performance bound that algorithms that use real models can be compared against, and that also provides an indication of the goodness of those models. We employ that performance bound in order to analytically evaluate a surrogate model assisted $(1 + 1)$-ES. Using the same bound, we also investigate the potential of performing incremental updates of Gaussian process surrogate models in an attempt to reduce algorithm internal computational costs and find that significant savings can be achieved at the cost of a small deterioration of model accuracy.

CCS CONCEPTS

• **Mathematics of computing** → **Bio-inspired optimization**; • **Computing methodologies** → **Continuous space search**.

KEYWORDS

Stochastic black-box optimization; evolution strategy; surrogate modelling

ACM Reference Format:
Hong Huang and Dirk V. Arnold. 2025. Surrogate Model Assisted Evolutionary Algorithms: Performance Bound and Incremental Gaussian Process Model Updates. In *Foundations of Genetic Algorithms XVIII (FOGA '25), August 27–29, 2025, Leiden, Netherlands.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3729878.3746619

1 INTRODUCTION

Surrogate models are in common use as a means to accelerate black-box optimization algorithms. Models are trained using information gained through the evaluation of the objective function in prior iterations, and they are used as possibly inaccurate but computationally relatively cheap surrogates for the true objective function. A wide variety of surrogate model assisted evolutionary algorithms have been proposed; see for example [2, 6, 12, 14, 19] and the references therein. Differences between algorithms include both choices with respect to how to use the surrogate models and the types of models employed. Regarding the former, algorithms may use surrogate models as an inexpensive means for determining whether points that have been sampled are worth evaluating using the objective function, or they may approximately optimize the surrogate models in order to determine where to sample next. Regarding the choice of models, low-rank polynomials, radial basis function regression, Gaussian processes, and ranking support vector machines have all been used.

Insights with regard to the abilities of surrogate model assisted evolutionary algorithms are commonly obtained empirically. A natural baseline for a comparison is the performance of the algorithm that the surrogate model assisted one is derived from, but that does not use the model. The effect of surrogate model assistance can then be quantified as a speed-up ratio: the number of objective function evaluations used by the unassisted algorithm to reach some termination criterion over the number used by the model assisted one.

We propose that another useful baseline to compare the performance of a surrogate model assisted evolutionary algorithm against is the performance of the algorithm using a hypothetical, perfect model; i.e., a model that accurately predicts the true objective function value without incurring the cost of an objective function evaluation. Rather than allowing to observe a speed-up gained from exploiting surrogate models, the proposed baseline provides a natural bound on the performance of the surrogate model assisted algorithm. The use of real surrogate models rather than hypothetical, perfect ones will generally result in a deterioration of performance. The benefits of the proposed baseline are twofold: first, observing the gap between the performance of the algorithm using the real model compared to that using the perfect model allows assessing the accuracy of the real model. And second, the assumption of perfect models potentially simplifies the algorithms to a point where a theoretical analysis of their performance becomes feasible.

In this paper we employ the proposed perfect model baseline in two contexts. First, we analyze the performance of the surrogate model assisted $(1 + 1)$-ES[1] with $(1, \lambda)$-preselection proposed by Yang and Arnold [20] assuming perfect models on sphere functions. The choice of algorithm is due to its simplicity, making it possible to use known results derived for model-free evolution strategies

[1] See Hansen et al. [7] for evolution strategy terminology.

FOGA '25, August 27–29, 2025, Leiden, Netherlands

ACM ISBN 979-8-4007-1859-5/2025/08…$15.00
https://doi.org/10.1145/3729878.3746619

in the analysis. And second, we consider the potential of incrementally updating Gaussian process surrogate models rather than performing regression from scratch whenever a new point is evaluated. Evaluating the algorithm using the perfect surrogate model baseline allows both judging the performance gap resulting from the use of imperfect models and the deterioration that results from incremental updates.

The remainder of this paper is organized as follows. Section 2 briefly discusses surrogate model assisted $(1 + 1)$-ES as well as Gaussian process surrogate models. Section 3 studies the $(1 + 1)$-ES with $(1, \lambda)$-preselection on sphere functions using the assumption of perfect models. Section 4 addresses the significant costs incurred in the training of Gaussian process surrogate models and studies the implications of the use of incremental updates. Section 5 concludes with a brief discussion and suggestions for future work.

2 BACKGROUND

Section 2.1 describes the surrogate model assisted $(1 + 1)$-ES introduced by Yang and Arnold [20], which is studied in Section 3 under the assumption of a perfect model. Section 2.2 briefly discusses Gaussian process surrogate models in preparation of the consideration of incremental updates in Section 4.

2.1 Surrogate Model Assisted (1+1)-ES

Yang and Arnold [20] propose a surrogate model assisted $(1 + 1)$-ES that extends the algorithm by Kayhani and Arnold [11] through the use of preselection. Both algorithms employ the surrogate model to determine whether or not to use the objective function $f : \mathbb{R}^n \to \mathbb{R}$ to evaluate a sampled point. Preselection forms an average of a set of randomly sampled points selected based on their surrogate model function values. Pseudo-code for a single iteration is presented in Algorithm 1. Parental candidate solution $\mathbf{x} \in \mathbb{R}^n$, step size parameter $\sigma > 0$, which is referred to as the mutation strength, and the set of previously evaluated points along with their objective function values determine the state of the strategy. Sample size parameters $\mu, \lambda \in \mathbb{N}$ with $\mu \leq \lambda$ are user supplied constants. The strategy in each iteration builds a surrogate model from the previously evaluated points, resulting in regression function $f_\epsilon : \mathbb{R}^n \to \mathbb{R}$, which serves as an approximation to the objective function. It then samples λ points from a normal distribution with mean $\mathbf{x}$, ranks them according to their surrogate model function values, and forms the average $\mathbf{y}$ of the μ best. That average is then evaluated using the surrogate model in order to determine whether or not to evaluate it using the true objective function. If the surrogate model suggests that $\mathbf{y}$ is inferior to $\mathbf{x}$, then the mutation strength is reduced and the iteration is complete. If the surrogate model suggests that $\mathbf{y}$ may be superior to the parental candidate solution, then it is evaluated using the objective function and the point joins the set of previously evaluated points. If $\mathbf{y}$ is inferior to $\mathbf{x}$ despite the surrogate model having suggested otherwise, then the mutation strength is reduced. Otherwise, $\mathbf{y}$ replaces $\mathbf{x}$ to form the parental candidate solution of the next iteration and the mutation strength is increased. Yang and Arnold [20] suggest settings $c_1 = 0.2$, $c_2 = c_3 = 1.0$, and $d = \sqrt{n + 1}$ for the parameters that control the rates of change of the step size parameter. Notice that at most one objective function evaluation is performed per iteration of the algorithm. Also notice that as surrogate model values are used only in comparisons, the assumption of perfect models for $(1 + 1)$-ES with $(\mu/\mu, \lambda)$-preselection can be relaxed to assuming that models accurately reflect the correct ordering of the λ points sampled.

Input:
- candidate solution $\mathbf{x} \in \mathbb{R}^n$
- mutation strength $\sigma \in \mathbb{R}_{>0}$
- a set of previously evaluated points with their objective function values

1: Build a surrogate model from previously evaluated points.
2: Independently sample λ trial step vectors $\mathbf{z}_k \in \mathbb{R}^n$, where $k = 1, 2, \ldots, \lambda$, from a multivariate standard normal distribution.
3: Evaluate the $\mathbf{y}_k = \mathbf{x} + \sigma\mathbf{z}_k$ using the surrogate model, yielding values $f_\epsilon(\mathbf{y}_k)$.
4: Compute

$$\mathbf{z} = \frac{1}{\mu} \sum_{j=1}^{\mu} \mathbf{z}_{j;\lambda} \,,$$

where $j;\lambda$ is the index of the jth smallest of the $f_\epsilon(\mathbf{y}_k)$.
5: Evaluate $\mathbf{y} = \mathbf{x} + \sigma\mathbf{z}$ using the surrogate model, yielding $f_\epsilon(\mathbf{y})$.
6: **if** $f_\epsilon(\mathbf{y}) \geq f(\mathbf{x})$ **then**
7: Let $\sigma \leftarrow \sigma\, \mathrm{e}^{-c_1/d}$.
8: **else**
9: Evaluate $\mathbf{y}$ using the objective function, yielding $f(\mathbf{y})$.
10: **if** $f(\mathbf{y}) \geq f(\mathbf{x})$ **then**
11: Let $\sigma \leftarrow \sigma\, \mathrm{e}^{-c_2/d}$.
12: **else**
13: Let $\mathbf{x} \leftarrow \mathbf{y}$ and $\sigma \leftarrow \sigma\, \mathrm{e}^{c_3/d}$.
14: **end if**
15: **end if**

Algorithm 1: Single iteration of the surrogate model assisted $(1 + 1)$-ES with $(\mu/\mu, \lambda)$-preselection.

Yang and Arnold [20] propose to analyze the algorithm by using unbiased Gaussian noise as a model for surrogate models. Perfect models as assumed below are included as noise of zero variance. However, their approach is moment based as it uses Gram-Charlier expansions [18] to approximate probability distributions. The quality of the approximation does not improve monotonically when including further moments. For perfect surrogate models the distributions being approximated are far from normal, and we find the accuracy of the approach inadequate. In Section 3 we therefore restrict ourselves to the case of $(1, \lambda)$-preselection (i.e., we assume $\mu = 1$) and pursue an approach that does not rely on moment based approximations.

2.2 Gaussian Process Models

Multiple surrogate model assisted evolutionary algorithms use Gaussian process regression [16] to generate surrogate models. While some, such as DTS-CMA-ES proposed by Bajer et al. [2] and akin to Bayesian optimization, use both the posterior mean and variance of the models, others, including GP-CMA-ES by Toal and Arnold [19], only use the mean.

Given a set $\{(\mathbf{x}_k, f(\mathbf{x}_k)), k = 1, 2, \ldots, m\}$ of m points $\mathbf{x}_k \in \mathbb{R}^n$ along with their function values $f(\mathbf{x}_k)$, a Gaussian process surrogate model is generated by forming $m \times m$ matrix $\mathbf{K}$ with entries $k_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$. We refer to that matrix as the kernel matrix. One of the most common choices for kernel function $\phi : \mathbb{R} \to \mathbb{R}$ is the squared exponential function with $\phi(d) = \exp(-(d/\theta)^2)$, where θ is a length scale parameter. The surrogate model function value of point $\mathbf{y} \in \mathbb{R}^n$ that is used in Lines 3 and 5 of Algorithm 1 is obtained by forming $m \times 1$ vector $\mathbf{k}$ with entries $k_i = \phi(\|\mathbf{y} - \mathbf{x}_i\|)$ and computing

$$f_\epsilon(\mathbf{y}) = f_0 + \mathbf{k}^\mathsf{T}\mathbf{K}^{-1}\mathbf{f}\,, \tag{1}$$

where $m \times 1$ vector $\mathbf{f}$ has entries $f_i = f(\mathbf{x}_i) - f_0$ for $i = 1, 2, \ldots, m$. Constant $f_0 \in \mathbb{R}$ implements a shift of the function values and in surrogate model assisted $(1 + 1)$-ES is set to the objective function value $f(\mathbf{x})$ of the current parental candidate solution. The length scale parameter θ of the Gaussian process model is set proportional to the value of the step size parameter σ of the algorithm. If θ is chosen too large, then the condition number of $\mathbf{K}$ increases to a point where accurate inversion becomes impossible for numerical reasons. If it is set too small, then f_0 unduly influences surrogate model function values.

While the number of objective function evaluations is often adopted as the measure of computational cost of surrogate model assisted evolutionary algorithms, algorithm internal costs (i.e., costs not related to the evaluation of the objective function) become considerable if the number of points used for generating surrogate models is large. Using large sets is desirable as it has the potential to improve the accuracy of the models. Toal and Arnold [19] begin generating surrogate models when $m = 2n$ points have been evaluated, and they limit the number of points used to generate a surrogate model to at most $m = (n + 2)^2$. Assuming that the computational costs of matrix inversion are cubic in the size of the matrix, the cost of generating a surrogate model using a number of points quadratic in the dimension are in $\Theta(n^6)$.

The significant cost of matrix operations in connection with covariance matrix adaptation evolution strategies (CMA-ES) [8, 9] has led to attempts to reduce algorithm internal computational costs. Igel et al. [10] use a Sherman-Morrison based update of the covariance matrix in $(1 + 1)$-CMA-ES, for which matrix updates are of rank one. In the case where matrix updates are of higher rank, Hansen [5] suggests to avoid performing the computationally expensive computation of eigen decompositions in every iteration, and to use inaccurate matrices in between those computations. If performed sufficiently frequently, the benefit is that amortized algorithm internal costs per iteration can be reduced from cubic to quadratic in the dimension while resulting in a negligible loss of performance. In Section 4 we study the potential of incremental updates of the inverse $\mathbf{K}^{-1}$ of the kernel matrix in an effort to the reduce the algorithm internal computational costs of Gaussian process surrogate modelling.

3 PERFECT MODEL PERFORMANCE BOUND

This section examines the performance on sphere functions of surrogate model assisted $(1 + 1)$-ES as described in Section 2.1 with $(1, \lambda)$-preselection and assuming a perfect surrogate model. Section 3.1 considers individual iterations. Section 3.2 investigates the step size adaptation component of the algorithm.

3.1 Single Iteration Behaviour

The sample points $\mathbf{y}_k$ of surrogate model assisted $(1 + 1)$-ES with $(1, \lambda)$-preselection are generated the same way as the offspring in simple $(1+\lambda)$-ES. Moreover, assuming a perfect model, the point selected as parent for the next iteration is the same for both strategies. What differs between the algorithm that employs surrogate models and the one that does not is their accounting of computational costs. While the $(1 + \lambda)$-ES performs λ objective function evaluations, the surrogate model assisted algorithm performs a single objective function evaluation if the best sample point is superior to the parent, and it performs none if it is not.

Expressions describing the single-step behaviour of $(1+\lambda)$-ES on sphere functions $f : \mathbb{R}^n \to \mathbb{R}$ with $f(\mathbf{x}) = g(\mathbf{x}^\mathsf{T}\mathbf{x})$, where function g is strictly increasing, have been derived by Beyer [3, 4]. Letting $\mathbf{x} \in \mathbb{R}^n$ denote the parent of the current iteration and $\mathbf{x}_\text{next} \in \mathbb{R}^n$ the point selected as the parent of the next iteration, and defining normalized mutation strength $\sigma^* = \sigma n/\|\mathbf{x}\|$ and normalized progress rate $\varphi^* = n\mathrm{E}[\|\mathbf{x}\| - \|\mathbf{x}_\text{next}\|]/\|\mathbf{x}\|$, Beyer derives the approximation

$$\varphi^* = \frac{\lambda\sigma^*}{\sqrt{2\pi}} \int_{\sigma^*/2}^{\infty} z\, \mathrm{e}^{-z^2/2}\, \Phi^{\lambda-1}(z)\, \mathrm{d}z - \frac{\sigma^{*2}}{2}\left[1 - \Phi^\lambda\left(\frac{\sigma^*}{2}\right)\right], \tag{2}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. That approximation is exact in the limit $n \to \infty$. Furthermore, he obtains the asymptotically exact expression

$$p_s = 1 - \Phi^\lambda\left(\frac{\sigma^*}{2}\right) \tag{3}$$

for the probability that the parent is replaced by one of the offspring. The normalized expected relative approach of the optimizer per unit of computational cost, which is measured in terms of numbers of objective function evaluations, for $(1 + \lambda)$-ES is thus

$$\eta_{1+\lambda} = \frac{\varphi^*}{\lambda}$$

and is referred to as the algorithm's efficiency. By determining the value of the mutation strength that maximizes $\eta_{1+\lambda}$, one finds that the fastest approach of the optimal solution per unit of computational cost can be achieved with $\lambda = 1$. The $(1 + 1)$-ES is the most efficient of all $(1 + \lambda)$-ES on sphere functions. Rechenberg [17] has derived the optimal normalized mutation strength $\hat{\sigma}^* = 1.224$ and the resulting value of $\hat{\eta}_{1+1} = 0.202$ for the maximal efficiency of that algorithm.

For $(1+1)$-ES with $(1, \lambda)$-preselection as described in Section 2.1, which perform a single objective function evaluation with probability p_s, from Eqs. (2) and (3), the normalized expected relative approach of the optimizer per unit of computational cost and thus their efficiency is

$$\begin{aligned}\eta_\text{pm} &= \frac{\varphi^*}{p_s} \\ &= \frac{\lambda\sigma^*}{\sqrt{2\pi}(1 - \Phi^\lambda(\sigma^*/2))} \int_{\sigma^*/2}^{\infty} z\, \mathrm{e}^{-z^2/2}\, \Phi^{\lambda-1}(z)\, \mathrm{d}z - \frac{\sigma^{*2}}{2}\,, \end{aligned} \tag{4}$$

where subscript 'pm' points to the assumption of a perfect model. Figure 1 illustrates the dependence of the normalized progress rate φ^*, the evaluation probability p_s, and the efficiency η_pm of

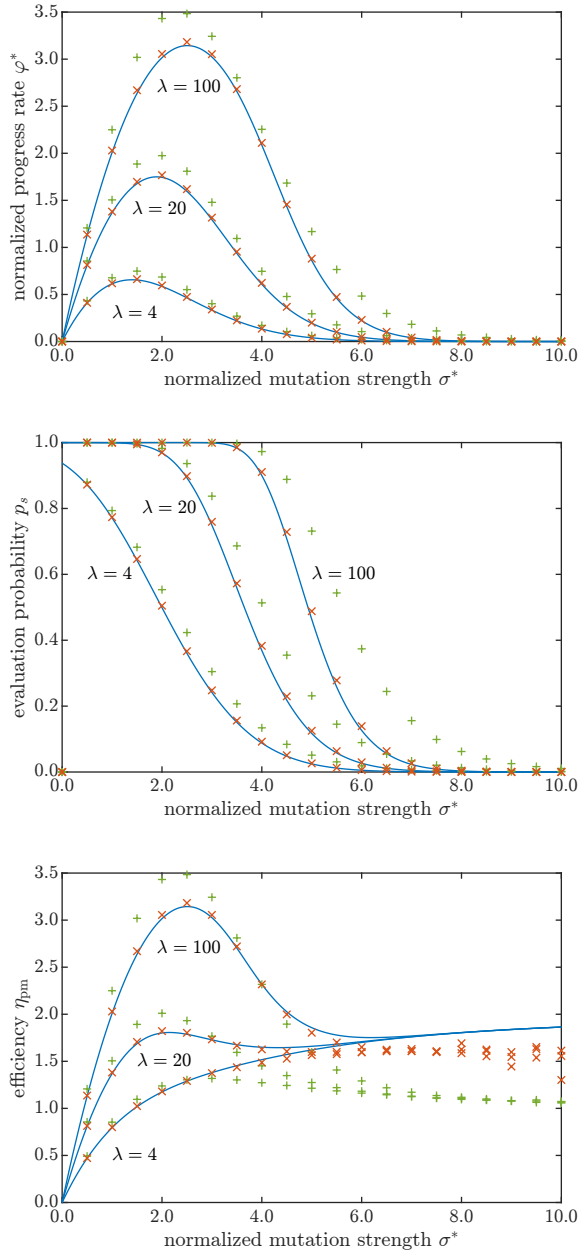

Figure 1: Normalized progress rate φ^*, evaluation probability p_s, and efficiency η_{pm} plotted against normalized mutation strength σ^*. The lines represent results from Eqs. (2), (3), and (4) for $\lambda = 4$, 20, and 100. The dots mark measurements obtained in computer experiments with $n = 10$ (green +) and $n = 100$ (red ×).

$(1+1)$-ES with $(1, \lambda)$-preselection on both the normalized mutation strength σ^* and the sample size λ. The curves have been generated using Eqs. (2), (3), and (4). From the top plot, the normalized progress rate is nonnegative and near zero for both small and large normalized mutation strengths, and it peaks at an intermediate value that depends on the sample size parameter λ. Normalized progress rates generally increase with increasing sample size. From the middle plot, the probability p_s that one of the offspring is superior to the parent and thus that an objective function evaluation is performed decreases with increasing mutation strength. Larger sample sizes generally increase evaluation probabilities. The bottom plot shows the efficiency of the algorithm, which incorporates both expected progress and expected cost. For small normalized mutation strengths, where evaluation probabilities for all but the smallest sample sizes are near 1.0, the efficiency closely mirrors the normalized progress rate. For larger normalized mutation strengths however, where evaluation probabilities markedly decrease, efficiency values significantly exceed normalized progress rates. While normalized progress rates for $\sigma^* \to \infty$ tend to zero, in the appendix we show that efficiencies η_{pm} tend to 2.0, irrespective of the sample size λ. For normalized mutation strengths larger than those that maximize the normalized progress rate, larger steps result in smaller expected progress per iteration, but simultaneously in reduced computational costs. The strategy that uses preselection based on a perfect model expends objective function evaluations only on those increasingly rare points that represent an improvement over the best solution found so far.

Each dot in Fig. 1 represents measurements made in 10^7 one-iteration experiments of $(1+1)$-ES with $(1, \lambda)$-preselection for $n \in \{10, 100\}$ and serves to illustrate the accuracy of the approximations that hold exactly only in the limit $n \to \infty$. Empirical progress rates have been obtained by averaging logarithmic progress values as described by Auger and Hansen [1]. It can be seen that Eqs. (2) and (3) appear visually accurate for $n = 100$ and underestimate both the normalized progress rates and the evaluation probabilities observed for $n = 10$. As a result of the underestimation of evaluation probabilities, which is present but less noticeable in the plots for $n = 100$, observed efficiencies differ markedly from predicted ones where normalized mutation strengths are high. Moreover, values for $n = 100$ with increasing σ^* appear increasingly noisy as a result of having been obtained from very few successful trials. The predicted evaluation probability at the right hand edge of the plots is below $3 \cdot 10^{-6}$, and the number of trials where one of the sampled points improves on the parent is thus small. Nonetheless, speed-ups relative to the maximum efficiency $\hat{\eta}_{1+1} = 0.202$ of $(1+1)$-ES that do not employ surrogate models are considerable.

By numerically evaluating Eq. (4), one observes that the efficiency η_{pm} of $(1+1)$-ES with $(1, \lambda)$-preselection and a perfect model for sample sizes $\lambda \leq 13$ monotonically increases with the normalized mutation strength and, as shown in the appendix, in the limit $\sigma^* \to \infty$ tends to a value of 2.0. For sample sizes $\lambda \geq 14$, the efficiency initially increases with increasing σ^*, then attains a local optimizer and begins decreasing before increasing again to ultimately approach the same limit value. Figure 2 illustrates the dependence of both the local maximizer and the local minimizer of the efficiency on the sample size. The lines have been generated by numerically optimizing Eq. (4) and then using Eq. (3) to obtain evaluation probabilities. From the left hand plot, both the minimizer and the maximizer of the efficiency are within narrow ranges for the sample sizes considered. From the middle plot, evaluation probabilities associated with the maximizer initially are about 0.8 and with increasing sample size rapidly reach values near 1.0. In

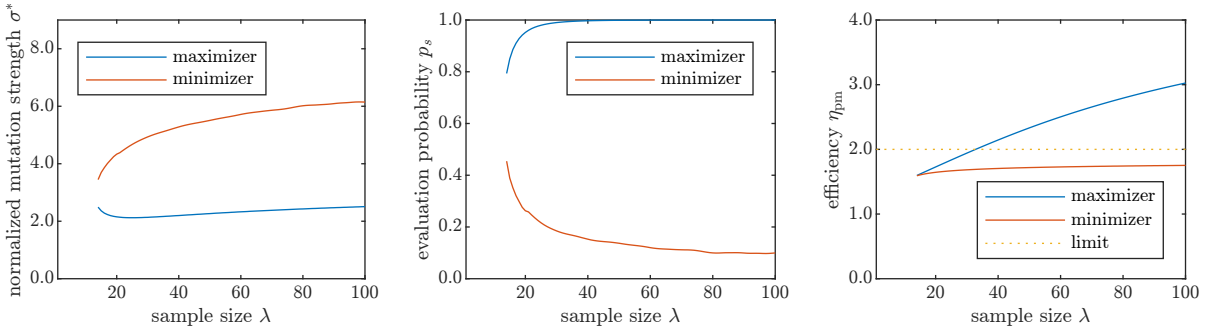

Figure 2: Normalized mutation strength σ^*, evaluation probability p_s, and efficiency η_{pm} of both the local maximizer (blue) and the local minimizer (red) of the efficiency plotted against sample size λ.

contrast, evaluation probabilities associated with the minimizer are never greater than 0.5 and decrease with increasing sample size. From the right hand plot, the efficiency of the local minimizer is below the limit value of 2.0, which is marked by a dotted line, by at most about twenty percent. The efficiency of the maximizer for sample sizes $\lambda \geq 33$ exceeds that limit value and grows with increasing sample size. Speed-up values associated with the maximizer relative to $(1+1)$-ES that do not employ surrogate models range from slightly below eight for $\lambda = 14$ to just under fifteen for $\lambda = 100$. From Fig. 1, predictions obtained in the limit $n \to \infty$ somewhat underestimate efficiencies associated with the local minimizer while they overestimate efficiencies obtained for large normalized mutation strengths.

3.2 Step Size Adaptation

With an appropriately controlled step size, $(1+1)$-ES with $(1,\lambda)$-preselection and perfect models on sphere functions linearly converge to the optimizer. The efficiency η_{pm} in that case is the dimension-normalized rate of convergence relative to the computational costs incurred by the algorithm. Rechenberg [17] refers to the range of mutation strengths that result in a substantial fraction of the optimal efficiency of an evolution strategy as the "evolution window". Whether or not the strategy is capable of operating within the evolution window depends on the performance of the step size adaptation component of the algorithm.

An approach to controlling the step size of evolution strategies is to set a target probability p_{target} for an observable event, and to either increase or decrease the mutation strength depending on whether the observed frequency of that event is above or below the target. For $(1+1)$-ES without surrogate model assistance, using the observable event of an offspring being superior to its parent and setting a target probability of twenty percent, the adaptation mechanism is known as the 1/5th-rule [17]. Akin to the implementation of that rule proposed by Kern et al. [13], $(1+1)$-ES with $(1,\lambda)$-preselection as given in Algorithm 1 and assuming perfect surrogate models effectively control their step size by establishing a target for the evaluation probability p_s. If none of the sampled points is superior to the parent, then no objective function evaluation is performed and $\log(\sigma)$ is decreased by c_1/d. If at least one of the sampled points is superior to the parent, then an objective function evaluation is performed and the

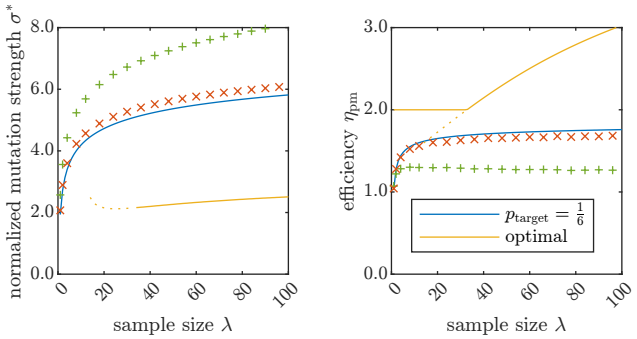

Figure 3: Normalized mutation strength σ^* and efficiency η_{pm} plotted against sample size λ. The blue lines mark predictions for target probability $p_{\text{target}} = \frac{1}{6}$. The solid and dotted yellow lines mark globally and locally optimal values, respectively. The dots mark measurements obtained in computer experiments with $n = 10$ (green +) and $n = 100$ (red ×).

logarithm of the mutation strength is increased by c_3/d. Thus, the logarithm of the mutation strength is unchanged in the mean if $(1-p_s)c_1 = p_s c_3$. Solving for p_s, coefficients c_1 and c_3 thus implicitly establish target probability $p_{\text{target}} = c_1/(c_1+c_3)$. For the parameter values proposed by Yang and Arnold [20], that target probability is $p_{\text{target}} = 0.2/(1.0{+}0.2) = \frac{1}{6}$. Parameter c_2 for a perfect model has no effect on the algorithm as the branch of the **if** statement in which it appears is never reached. Similar to the shrinking of the trust-region in a trust-region method, in connection with imperfect models it has the effect of reducing the step size if the model is found to be inaccurate.

Figure 3 illustrates normalized mutation strengths generated and efficiencies achieved by the algorithm. The blue lines have been obtained by solving Eq. (3) for the normalized mutation strength that results in $p_s = p_{\text{target}} = \frac{1}{6}$ and then using that mutation strength in Eq. (4) to compute the corresponding efficiencies. The solid yellow lines mark optimal normalized mutation strengths and efficiencies as determined in Section 3.1. For sample sizes $\lambda < 33$ globally optimal efficiencies are approached as $\sigma^* \to \infty$ and dotted yellow lines illustrate the locally optimal values obtained above. Even though the target probability is far from its optimal value (near

zero for $\lambda < 33$ and near one for larger sample sizes), efficiencies that are obtained with $p_{\text{target}} = \frac{1}{6}$ are within a factor of two of optimal for all sample sizes considered. The reason for this is evident from Figs. 1 and 2. For small sample sizes, the target evaluation rate is sufficiently small to result in mutation strengths large enough to realize a substantial fraction of the limit value achieved for $\sigma^* \to \infty$. For larger sample sizes, the target evaluation rate commonly puts the strategy in the vicinity of the efficiency minimizer. However, the efficiency at that minimizer is a substantial fraction of the maximal efficiency unless the sample size is very large. Step size adaptation is successful as, in contrast to $(1+\lambda)$-ES, where too large a mutation strength results in significant computational cost while likely not allowing progress toward the optimizer, sampling only unsuccessful points when using perfect surrogate models does not trigger any objective function evaluations. Surrogate models make the task of step size adaptation easier as they result in a far wider evolution window. The shallow slope of the blue line in the right hand plot in Fig. 3 also suggests that due to imperfectly adapted mutation strengths, the benefit from increasing sample sizes beyond a value of about twenty is very moderate. The green and red points mark data that have been measured in runs of Algorithm 1 assuming perfect surrogate models in dimensions $n \in \{10, 100\}$ and illustrate that while the results derived in the limit $n \to \infty$ and that assume instantaneous adaptation (in that the target probability is assumed to always be attained rather than aspired to be reached in a gradual process) are not perfectly accurate, they do capture the behaviour of the algorithm quite well. Speed-up factors compared to $(1+1)$-ES that do not use surrogate models commonly are between six and eight.

While the results thus derived can serve as an approximate bound on the performance that surrogate model assisted $(1 + 1)$-ES with $(1, \lambda)$-preselection can achieve on sphere functions, a question that remains open is how much real surrogate models differ from the perfect ones assumed in the analysis. The following section, in addition to studying the impact of incremental updates, addresses that question experimentally for a wider range of objective functions.

4 INCREMENTAL MODEL UPDATE

This section considers $(1+1)$-ES employing Gaussian process surrogate models. Its objectives are to explore the potential of incremental updates of the models to reduce algorithm internal costs, and to evaluate the accuracy of Gaussian process models relative to the perfect model baseline. Section 4.1 details the algorithmic modifications needed to incrementally update Gaussian process surrogate models. Section 4.2 describes initial experiments to determine useful parameter settings, and Section 4.3 experimentally evaluates the impact of incremental updates on the number of objective function evaluations as well as on algorithm internal costs.

4.1 Algorithmic Modifications

In those iterations where the surrogate model assisted $(1 + 1)$-ES chooses to use the objective function to evaluate a candidate solution, the surrogate model needs to be updated. A deferred update, similar to the use of deferred eigen decompositions in CMA-ES, is not a viable approach as the information gained from evaluating the most recent point likely is highly relevant for accurately estimating the function values of future points. Instead, we employ blockwise updates [15] of the inverse kernel matrix in order to reduce computational cost. Adding a point to the training set amounts to adding a row and column to $m \times m$ kernel matrix $\mathbf{K}$. Letting $\mathbf{c}$ denote an $m \times 1$ column vector, $\mathbf{r}$ a $1 \times m$ row vector, and $e \in \mathbb{R}$, the inverse of the expanded matrix is

$$\begin{bmatrix} \mathbf{K} & \mathbf{c} \\ \mathbf{r} & e \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{K}^{-1} + \frac{1}{k}(\mathbf{K}^{-1}\mathbf{c})(\mathbf{r}\mathbf{K}^{-1}) & -\frac{1}{k}\mathbf{K}^{-1}\mathbf{c} \\ -\frac{1}{k}\mathbf{r}\mathbf{K}^{-1} & \frac{1}{k} \end{bmatrix},$$

where $k = (e - \mathbf{r}\mathbf{K}^{-1}\mathbf{c}) \in \mathbb{R}$ is the Schur complement of $\mathbf{K}$. Notice that by grouping the matrix-vector products as indicated by the brackets, the computation of the expanded inverse can be accomplished in time in $O(m^2)$. Similarly, removing a point from the training set can be accomplished by removing a row and column from matrix $\mathbf{K}$. As the surrogate model assisted $(1 + 1)$-ES always removes the oldest point first, it is the top row and left most column that need to be removed. If the inverse of the original kernel matrix is

$$\begin{bmatrix} e & \mathbf{r} \\ \mathbf{c} & \mathbf{K} \end{bmatrix}^{-1} = \begin{bmatrix} u & \mathbf{v} \\ \mathbf{w} & \mathbf{M} \end{bmatrix},$$

where $u \in \mathbb{R}$, $\mathbf{v}$ is a $1 \times m$ row vector, and $\mathbf{w}$ is an $m \times 1$ column vector, then the inverse of the reduced kernel matrix is

$$\mathbf{K}^{-1} = \mathbf{M} - \frac{\mathbf{w}\mathbf{v}}{u}.$$

Notice that this calculation, too, can be accomplished in time in $O(m^2)$. The incremental update of $\mathbf{K}^{-1}$ thus allows making updates of the surrogate model in an amount of time that is quadratic in the size of the training set.

There are three issues that arise when employing incremental updates of the inverse kernel matrix in the surrogate model assisted $(1 + 1)$-ES. First, for the squared exponential kernel and the setting of the length scale parameter θ recommended by Yang and Arnold [20], condition numbers of the kernel matrix may commonly be very large. As a result, due to the limited numerical accuracy of floating point representations, the accuracy of the matrix inverse decreases with an increasing number of incremental updates and the performance of the algorithm gradually deteriorates. We thus perform a full matrix inversion (at cubic cost) after κn new points have been evaluated, with incremental updates in between. The amortized per-iteration cost of surrogate modelling is thus in $O(m^2 + m^3/(\kappa n))$. Second, surrogate model assisted $(1 + 1)$-ES as described above in each iteration set the length scale parameter θ to a value proportional to the step size parameter σ, thus preventing an iterative update of the inverse kernel matrix. We set θ proportional to σ only on those iterations where a full inverse of the kernel matrix is computed, and we keep it constant in between. We have not observed any negative consequences from this. And third, inaccurate surrogate model predictions make it advisable to replace condition $f_\epsilon(\mathbf{y}) \geq f(\mathbf{x})$ in Line 6 of Algorithm 1 with condition $f_\epsilon(\mathbf{y}) \geq f_\epsilon(\mathbf{x})$.

4.2 Parameter Setting

It remains to find a useful setting for parameter κ, which determines the frequency with which full matrix inverses are computed and which impacts both the performance of the algorithm in terms of the number of objective function evaluations (as less frequent updates

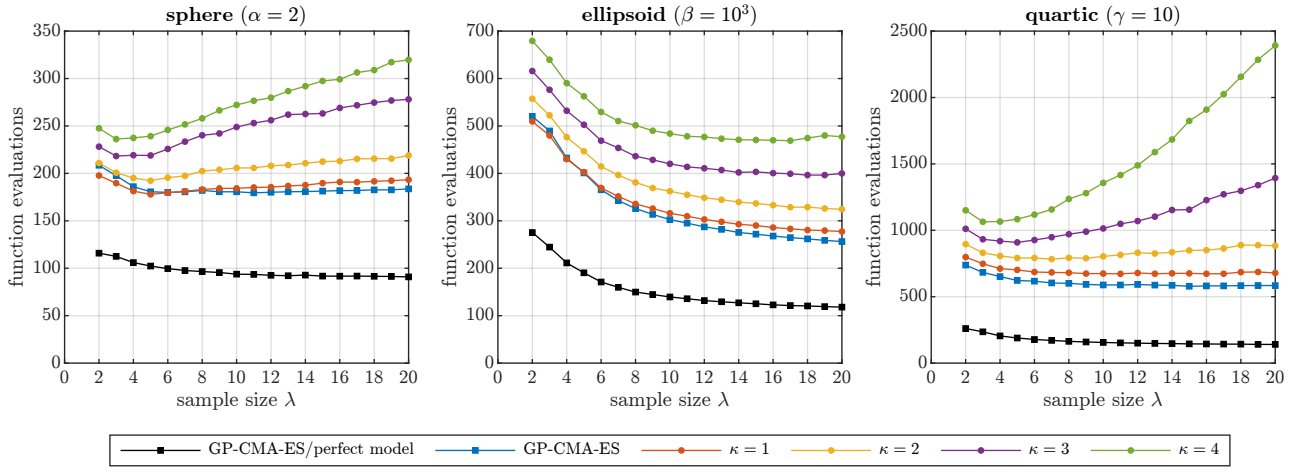

Figure 4: Median numbers of objective function evaluations required to solve sphere, ellipsoid, and quartic functions plotted against the sample size λ. The curves labelled GP-CMA-ES are for the algorithm that performs a full matrix inversion in each iteration, those labelled GP-CMA-ES/perfect model for a variant of the algorithm that uses a hypothetical, perfect model instead of a Gaussian process model. The remaining curves are for the algorithm using Gaussian process models, but with a full matrix inversion performed only every κn iterations, with incremental updates in between.

result in less accurate surrogate models) and algorithm internal computational costs. In order to explore the trade-off, we have implemented incremental surrogate model updates as described above in GP-CMA-ES, which add covariance matrix adaptation to $(1+1)$-ES as considered above and which employ preselection with weighted recombination. The algorithm as well as all parameter settings are taken from Toal and Arnold [19]. As that reference, we consider three families of test functions:

- Sphere functions $f(\mathbf{x}) = (\mathbf{x}^\mathsf{T}\mathbf{x})^{\alpha/2}$. For $\alpha = 2$ this family includes the quadratic sphere. The optimizer is at zero.
- Convex quadratic functions $f(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{B}\mathbf{x}$ where the symmetric $n \times n$ matrix $\mathbf{B}$ has eigenvalues $b_{ii} = \beta^{(i-1)/(n-1)}$, $i = 1, \ldots, n$, with condition number $\beta \geq 1$. The quadratic sphere is included as the special case with $\beta = 1$. We refer to this family as ellipsoid functions. The optimizer is at zero.
- Quartic functions $f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[\gamma(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right]$. For $\gamma = 100$ this family includes the generalized Rosenbrock function. The global optimizer is at $(1, 1, \ldots, 1)^\mathsf{T}$. A second, merely local minimizer exists for $n \geq 4$.

All runs are initialized by sampling random starting points uniformly in $[-4, 4]^n$. The step size parameter is initialized to $\sigma = 10$. Runs are terminated when a point $\mathbf{x}$ with $f(\mathbf{x}) < 10^{-8}$ has been generated and evaluated. Runs on the quartic functions that converge to the merely local optimizer are removed from consideration.

Figure 4 shows median numbers of objective function evaluations required to reach termination accuracy across one thousand independent runs on the quadratic sphere, the ellipsoid with condition number 10^3 of its Hessian matrix, and the quartic function with parameter $\gamma = 10$. In all cases, the dimension is $n = 10$. The algorithm represented in blue is a GP-CMA-ES that computes full matrix updates in each iteration. The data points represented in black are for a variant of that algorithm that uses hypothetical, perfect models instead of Gaussian process models. The remaining curves are for variants of the Gaussian process model assisted algorithm that perform full inversions only after κn new points have been evaluated for $\kappa \in \{1, 2, 3, 4\}$, with incremental updates in between. As expected, the algorithm that performs full matrix inversions in each iteration requires smaller numbers of objective function evaluations compared to those algorithm variants that rely on incremental updates. The performance of the algorithm deteriorates gradually with increasing κ as a result of inaccurate surrogate models. As for $\kappa > 2$ and larger sample sizes a very significant increase in the number of objective function evaluations on the quartic function is observed, we use $\kappa = 1$ in all of what follows. That choice does not increase the number of objective function evaluations on any of the problems considered by more than twenty percent. Greater amounts of deterioration may be observed with sample sizes larger than those represented in the figure (as the algorithm more intensively exploits potentially inaccurate surrogate models), but the results from Section 3 suggest that there may be little benefit to be derived from larger sample sizes even when using accurate models.

4.3 Evaluation

In order to evaluate the potential of incremental surrogate model updates, we consider the same three families of test problems as above, but we explore larger ranges for both the parameters of the families and the dimensions considered. A complication when measuring speed-ups that result from incremental updates is that for matrices above a certain size, MATLAB employs multiple threads for matrix inversion, thus resulting in a speed-up from using parallel computing resources that is by default not available when performing incremental updates. As the availability of parallel hardware resources is machine dependent, for the purpose of performance

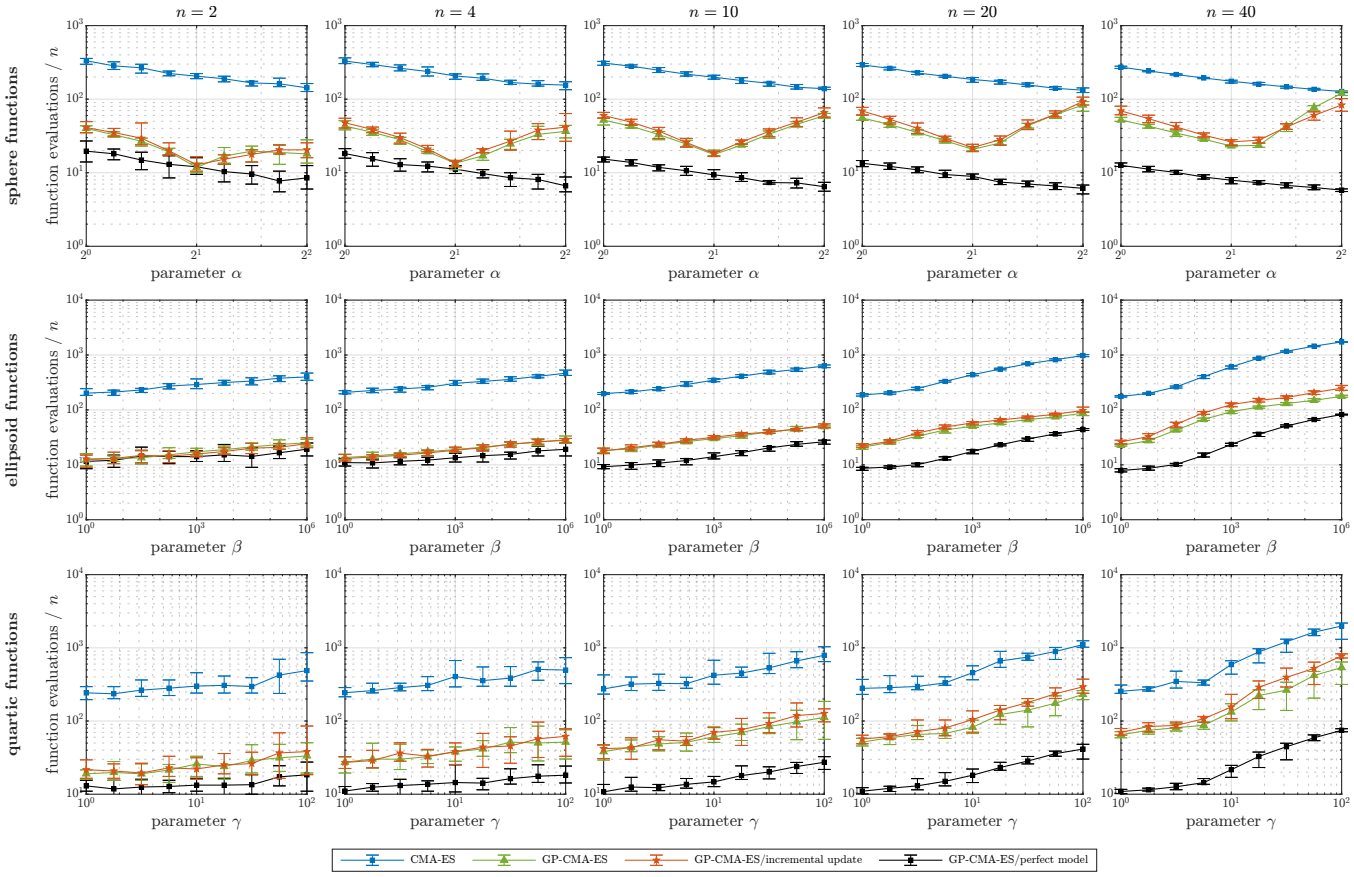

Figure 5: Numbers of objective function evaluations per dimension required to attain termination accuracy on sphere functions, ellipsoid functions, and quartic functions plotted against the parameter parameterizing each of the test function families. The algorithms shown are $(\mu/\mu, \lambda)$-CMA-ES without surrogate model assistance (blue), surrogate model assisted CMA-ES with $(\mu/\mu, \lambda)$-preselection and a hypothetical, perfect surrogate model that predicts exact function values (black), GP-CMA-ES as proposed by Toal and Arnold [19] (green), and our variant of the latter algorithm that performs incremental surrogate model updates (red).

measurement we choose to remove MATLAB's ability to utilize multiple threads and run all code in a single thread. A practically more useful solution of course is to instead allow for multiple threads for the incremental updates, which are easily parallelized.

Figure 5 shows the number of objective function evaluations per dimension required to reach termination accuracy for all test problem instances considered, with problem dimensions ranging from $n = 2$ to 40. All algorithms use sample size $\lambda = 10$, and errorbars represent the minimum, mean, and maximum values observed across eleven independent runs. Results for $(\mu/\mu, \lambda)$-CMA-ES without surrogate model assistance, which require the largest numbers of function evaluations, are represented in blue. Results obtained using GP-CMA-ES as described by Toal and Arnold [19] are shown in green. Data represented in black are for the surrogate model assisted algorithm with a hypothetical, perfect surrogate model as proposed above. The data shown in red represent GP-CMA-ES using incremental updates, with full updates performed only once n new points have been evaluated. The gaps between the green and black data points suggest that the Gaussian process surrogate models are near perfect for those problems that are quadratic and of low dimension, and that both increasing dimension and increasing deviation from being quadratic result in a deterioration of model quality. The relatively narrow gaps between the red and the green data points show that the setting of κ derived from a small number of test instances in Section 4.2 does not result in more than a moderate loss of performance stemming from the use of incremental updates for any of the problem instances considered. The gaps appear most significant on quartic functions with large γ for dimensions $n \geq 20$, where the algorithm that relies on incremental updates of the inverse kernel matrix requires numbers of objective function evaluations about twenty percent larger than those required by the non-incremental algorithm.

Finally, Figure 6 compares the running times of the surrogate model related components of the algorithms by presenting the speed-up resulting from the use of incremental updates of the inverse kernel matrix relative to performing a full inversion of

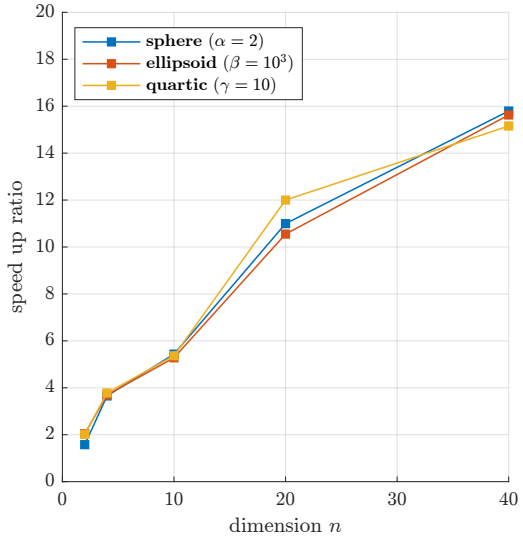

Figure 6: Speed-up resulting from the use of incremental updates of the inverse kernel matrix plotted against problem dimension n.

that matrix in each iteration. The results shown are for quadratic spheres, ellipsoids with condition number $\beta = 10^3$ of their Hessian, and quartic functions with parameter $\gamma = 10$. All data points represent median values from eleven runs. It can be seen that the incremental updates result in a speed-up that is approximately linear in the dimension of the problems, with a speed-up factor of nearly sixteen for $n = 40$. If overall running times are not strongly dominated by the cost of objective function evaluations, then that speed-up can be considerable.

5 CONCLUSIONS

To conclude, we have proposed to consider hypothetical, perfect surrogate models in an effort to better understand surrogate model assisted evolutionary algorithms. Assuming perfect surrogate models allows placing a bound on the speed-up that a surrogate model assisted evolutionary algorithm may achieve at best, and it also allows judging the accuracy of real surrogate models. We have presented an analysis of a perfect surrogate model assisted $(1+1)$-ES on sphere functions that shows how the efficiency of the algorithm scales with increasing sample size. We have then explored the potential of incremental updates of Gaussian process surrogate models and found that periodic refreshes are needed to prevent the loss of accuracy of the models from unduly impacting the performance of the algorithms. By comparing with the perfect model baseline we have found that significantly reduced algorithm internal computational costs can be achieved at the cost of a very moderate increase in the number of objective function evaluations. We have also been able to judge the ability of Gaussian process surrogate models with a quadratic exponential kernel to model different test functions.

In future work, we plan to extend the analysis in Section 3 to $(\mu/\mu, \lambda)$-preselection with $\mu > 1$. We will also use the assumption of perfect surrogate models to analyze the performance of other surrogate model assisted evolutionary algorithms on simple test functions, opening up the possibility of comparing algorithms separately from the surrogate models that they use. Finally, we will explore implications of the choice of models and their kernel functions in those cases where the gap observed between perfect surrogate models and real surrogate models is large.

ACKNOWLEDGEMENTS

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

APPENDIX

The efficiency of the surrogate model assisted $(1+1)$-ES with $(1, \lambda)$-preselection on sphere functions in the limit $n \to \infty$ is described by Eq. (4), which, after rearranging terms, can be written as

$$\eta_{\text{pm}} = \frac{\sigma^{*2}}{2}(AB - 1)\,,$$

where

$$A = \frac{\lambda\, \mathrm{e}^{-\sigma^{*2}/8}}{\sqrt{2\pi}(1 - \Phi^{\lambda}(\sigma^*/2))\sigma^*/2}$$

and

$$B = \mathrm{e}^{\sigma^{*2}/8} \int_{\sigma^*/2}^{\infty} z\, \mathrm{e}^{-z^2/2}\, \Phi^{\lambda-1}(z)\, \mathrm{d}z\,.$$

The limit behaviour for large normalized mutation strengths can be obtained by utilizing Mill's ratio [18] to obtain asymptotic expansion

$$\Phi(z) = 1 - \frac{1}{\sqrt{2\pi}z}\, \mathrm{e}^{-z^2/2} \left(1 - \frac{1}{z^2} + O\left(\frac{1}{z^4}\right)\right)$$

of the tail of the standard normal distribution. Letting $z = \sigma^*/2$ and using the binomial theorem it follows that

$$1 - \Phi^{\lambda}\left(\frac{\sigma^*}{2}\right) = \frac{\lambda}{\sqrt{2\pi}\sigma^*/2}\, \mathrm{e}^{-\sigma^{*2}/8} \left(1 - \frac{4}{\sigma^{*2}} + O\left(\frac{1}{\sigma^{*4}}\right)\right)$$

and thus that

$$A = 1 + \frac{4}{\sigma^{*2}} + O\left(\frac{1}{\sigma^{*4}}\right)\,. \tag{5}$$

As $0 < \Phi(z) < 1$ for all $z \in \mathbb{R}$ and $\Phi(z)$ is strictly increasing with z, B can be bounded as

$$\begin{aligned}\Phi^{\lambda}(\sigma^*/2) < \Phi^{\lambda-1}(\sigma^*/2)\, \mathrm{e}^{\sigma^{*2}/8} \int_{\sigma^*/2}^{\infty} z\, \mathrm{e}^{-z^2/2}\, \mathrm{d}z \\ < B < \mathrm{e}^{\sigma^{*2}/8} \int_{\sigma^*/2}^{\infty} z\, \mathrm{e}^{-z^2/2}\, \mathrm{d}z = 1\end{aligned}$$

and thus, again using Mill's ratio,

$$B = 1 + O\left(\frac{\mathrm{e}^{-\sigma^{*2}/8}}{\sigma^*}\right)\,. \tag{6}$$

Combining the expressions for A and B in Eqs. (5) and (6) yields

$$\begin{aligned}\eta_{\text{pm}} &= \frac{\sigma^{*2}}{2}\left[\left(1 + \frac{4}{\sigma^{*2}} + O\left(\frac{1}{\sigma^{*4}}\right)\right)\left(1 + O\left(\frac{\mathrm{e}^{-\sigma^{*2}/8}}{\sigma^*}\right)\right) - 1\right] \\ &= 2 + O\left(\frac{1}{\sigma^{*2}}\right)\,,\end{aligned}$$

thus showing that $\lim_{\sigma^* \to \infty} \eta_{\text{pm}} = 2$.

REFERENCES

[1] A. Auger and N. Hansen. 2006. Reconsidering the progress rate theory for evolution strategies in finite dimensions. In *Genetic and Evolutionary Computation Conference — GECCO 2006*. ACM Press, 445–452. https://doi.org/10.1145/1143997.1144081

[2] L. Bajer, Z. Pitra, J. Repický, and M. Holeňa. 2019. Gaussian process surrogate models for the CMA evolution strategy. *Evolutionary Computation* 27, 4 (2019), 665–697. https://doi.org/10.1162/evco_a_00244

[3] H.-G. Beyer. 1993. Toward a theory of evolution strategies: Some asymptotical results from the $(1 \stackrel{+}{,} \lambda)$-theory. *Evolutionary Computation* 1, 2 (1993), 165–188. https://doi.org/10.1162/evco.1993.1.2.165

[4] H.-G. Beyer. 2001. *The Theory of Evolution Strategies*. Springer Verlag. https://doi.org/10.1007/978-3-662-04378-3

[5] N. Hansen. 2016. The CMA evolution strategy: A tutorial. arxiv:1604.00772. https://doi.org/10.1007/978-3-662-43505-2_44

[6] N. Hansen. 2019. A global surrogate assisted CMA-ES. In *Genetic and Evolutionary Computation Conference — GECCO 2019*. ACM Press, 664–672. https://doi.org/10.1145/3321707.3321842

[7] N. Hansen, D. V. Arnold, and A. Auger. 2015. Evolution strategies. In *Springer Handbook of Computational Intelligence*, J. Kacprzyk and W. Pedrycz (Eds.). Springer Verlag, Berlin, Heidelberg, 871–898. https://doi.org/10.1007/978-3-662-43505-2_44

[8] N. Hansen, S. D. Müller, and P. Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation* 11, 1 (2003), 1–18. https://doi.org/10.1162/106365603321828970

[9] N. Hansen and A. Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9, 2 (2001), 159–195. https://doi.org/10.1162/106365601750190398

[10] C. Igel, T. Suttorp, and N. Hansen. 2006. A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. ACM Press, Seattle, WA, 453–460. https://doi.org/10.1145/1143997.1144082

[11] A. Kayhani and D. V. Arnold. 2018. Design of a surrogate model assisted $(1 + 1)$-ES. In *Parallel Problem Solving from Nature — PPSN XV*, A. Auger et al. (Eds.). Springer Verlag, 16–28. https://doi.org/10.1007/978-3-319-99253-2_2

[12] S. Kern, N. Hansen, and P. Koumoutsakos. 2006. Local meta-models for optimization using evolution strategies. In *Parallel Problem Solving from Nature — PPSN IX*, T. P. Runarsson et al. (Eds.). Springer Verlag, 939–948. https://doi.org/10.1007/11844297_95

[13] S. Kern, S. D. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. 2004. Learning probability distributions in continuous evolutionary algorithms — A comparative review. *Natural Computing* 3, 1 (2004), 77–112. https://doi.org/10.1023/B:NACO.0000023416.59689.4e

[14] I. Loshchilov, M. Schoenauer, and M. Sebag. 2013. Intensive surrogate model exploitation in self-adaptive surrogate-assisted CMA-ES. In *Genetic and Evolutionary Computation Conference — GECCO 2013*. ACM Press, 439–446. https://doi.org/10.1145/2463372.2463427

[15] T.-T. Lu and S.-H. Shiou. 2002. Inverses of 2×2 block matrices. *Computers & Mathematics with Applications* 43, 1 (2002), 119–129. https://doi.org/10.1016/S0898-1221(01)00278-4

[16] C. E. Rasmussen and C. K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press. https://doi.org/10.7551/mitpress/3206.003.0019

[17] I. Rechenberg. 1973. *Evolutionsstrategie — Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog.

[18] A. Stuart and J. K. Ord. 1994. *Kendall's Advanced Theory of Statistics* (sixth ed.). Vol. I: Distribution Theory. Wiley.

[19] L. Toal and D. V. Arnold. 2020. Simple surrogate model assisted optimization with covariance matrix adaptation. In *Parallel Problem Solving from Nature — PPSN XVI*, T. Bäck et al. (Eds.). Springer Verlag, 1–14. https://doi.org/10.1007/978-3-030-58112-1_13

[20] J. Yang and D. V. Arnold. 2019. A Surrogate Model Assisted (1+1)-ES with Increased Exploitation of the Model. In *Proceedings of the 2019 Genetic and Evolutionary Computation Conference — GECCO 2019*. ACM Press, 727–735. https://doi.org/10.1145/3321707.3321728