Gaussian Adaptation as a unifying framework for continuous black-box optimization and adaptive Monte Carlo sampling

Christian L. Müller, Ivo F. Sbalzarini

Abstract— **We present a unifying framework for continuous optimization and sampling. This framework is based on Gaussian Adaptation (GaA), a search heuristic developed in the late 1960's. It is a maximum-entropy method that shares several features with the (1+1)-variant of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The algorithm samples single candidate solutions from a multivariate normal distribution and continuously adapts the first and second moments. We present modifications that turn the algorithm into both a robust continuous black-box optimizer and, alternatively, an adaptive Random Walk Monte Carlo sampler. In black-box optimization, sample-point selection is controlled by a monotonically decreasing, fitness-dependent acceptance threshold. We provide general strategy parameter settings, stopping criteria, and restart mechanisms that render GaA *quasi* parameter free. We also introduce Metropolis GaA (M-GaA), where sample-point selection is based on the Metropolis acceptance criterion. This turns GaA into a Monte Carlo sampler that is conceptually similar to the seminal Adaptive Proposal (AP) algorithm. We evaluate the performance of Restart GaA on the CEC 2005 benchmark suite. Moreover, we compare the efficacy of M-GaA to that of the Metropolis-Hastings and AP algorithms on selected target distributions.**

I. Introduction

A large class of problems in science and engineering can be formulated as global optimization problems or as sampling problems. Global optimization is concerned with finding a single or a set of optimal solutions for a given problem specification. Sampling consists of correctly drawing random samples from a given probability distribution. In many cases, optimization and sampling algorithms have to operate in a black-box scenario, where only zeroth-order information about the objective function or the target probability distribution is available. In black-box optimization, only objective function values can be obtained. Analytical gradients or Hessians are not available, or do not exist. Many practical applications, including parameter estimation in electrical or biological networks, are of this kind. Indirect (or black-box) sampling is used when the target probability distribution is not explicitly known, or is known only up to a normalizing constant. This is often the case in Bayesian statistics and in statistical physics, where the unknown normalization constant is given by the partition function of the state space.

For both problem classes, Monte Carlo methods have become the prevalent computational paradigm. They rely on iterative random sampling in order to approximate the desired result. A crucial design decision is *how* the random samples are generated. In continuous spaces, multivariate Gaussian distributions are the standard choice. Several continuous black-box optimization methods, such as Simulated Annealing (SA) in general state spaces [1], Gaussian Adaptation (GaA) [2], and Evolution Strategies (ES) use Gaussian sampling to generate candidate solutions. For indirect sampling, Green and Han [3] were among the first to employ Gaussian distributions. In order to sample from a specific target distribution, their algorithm draws random variates from a Gaussian distribution and evaluates the target distribution at these sample points. A specific acceptance-rejection scheme, proposed by Metropolis et al. [4], guarantees that the process follows the desired target distribution. Methods of this type are generally referred to as Markov Chain Monte Carlo (MCMC) methods.

Both the first ES, Rechenberg's (1+1)-ES, and the standard Random Walk Metropolis sampling algorithm [5] use single samples from an *isotropic* multivariate Gaussian distribution. More recent algorithms constantly adapt the covariance matrix of the sampling distribution according to previously accepted samples. This includes optimization algorithms such as Hansen's ES with Covariance Matrix Adaptation (CMA-ES) [6] and Kjellström's GaA algorithm [2], [7]. An important conceptual difference between CMA-ES and GaA is the purpose of covariance adaptation: While CMA-ES is designed to increase the likelihood of generating successful *mutations*, GaA adapts the covariance such as to maximize the *entropy* of the search distribution under the constraint that acceptable search points are found with a predefined, fixed hitting probability. Covariance matrix adaptation is also used in indirect sampling. Haario et al. [8] remedied the well-known inefficiency of the Metropolis algorithm on high-dimensional and/or highly distorted target distributions by continuously adapting the Gaussian proposal distribution. They thus introduced the seminal Adaptive Proposal (AP) algorithm [8] based on covariance matrix adaptation. The AP algorithm has been empirically shown to outperform the classical Metropolis algorithm, yet at the expense of sacrificing rigorous convergence proofs for general target distributions.

Here, we present a unifying formulation for continuous black-box optimization and adaptive Monte Carlo sampling based on GaA. We first revisit the key concepts of GaA and its relation to ES. We suggest general parameter settings, convergence criteria, and a restart mechanism for GaA. The resulting Restart GaA is a quasi parameter-free off-the-shelf black-box optimizer. We benchmark Restart GaA on the

C. L. Müller and I. F. Sbalzarini are with the Institute of Theoretical Computer Science and the Swiss Institute of Bioinformatics, ETH Zurich, CH–8092 Zürich, Switzerland (phone: +41-44-6325512, +41-44-6326344; fax: +41-44-6321562; e-mail: christian.mueller@inf.ethz.ch, ivos@ethz.ch).

full set of the IEEE CEC 2005 test suite, and we provide guidelines when to use Restart GaA in practice. We then introduce Metropolis' acceptance-rejection scheme as selection mechanism in GaA and show that this modification turns GaA into a Metropolis algorithm with adaptive proposal (M-GaA). We highlight the similarities and differences between M-GaA and the AP algorithm, and we assess the performance of M-GaA on benchmark target distributions.

II. Gaussian Adaptation

We summarize the key concepts of the canonical GaA algorithm as developed by Kjellström and co-workers. We then propose a standard parametrization, constraint handling, convergence criteria, and a restart strategy, resulting in the Restart GaA algorithm. We further introduce M-GaA as an adaptive sampling algorithm based on GaA.

A. Canonical Gaussian Adaptation

GaA has been developed in the context of analog circuit design. There, one key objective is to find optimal values for certain design parameters $\mathbf{x} \in \mathcal{A} \subset \mathbb{R}^n$, e.g. nominal values of resistors and capacitors, that fulfill two requirements: First, the parameter values should satisfy some (real-valued) objective (or criterion) function $f(\mathbf{x})$ applied to the circuit output. Second, the nominal values should be robust with respect to intrinsic random variations of the components during operation. Kjellström noticed that with increasing network complexity stochastic methods that only rely on evaluations of the objective function are superior to classical optimization schemes. Starting from an exploration method that can be considered an adaptive random walk through parameter space [9], he refined his algorithm to what he called Gaussian Adaptation [2].

In order to develop a search heuristic that satisfies both design requirements, Kjellström implicitly applied the maximum-entropy principle [10]. This fundamental principle of statistical inference leads to the least biased estimate possible on the given (incomplete) information. In the case of given mean and covariance information, the Gaussian distribution maximizes the entropy $\mathcal{H}$, and hence is the preferred (i.e. least biased) choice to search and characterize the space of acceptable points. The entropy of a multivariate Gaussian distribution $\mathcal{N}(\mu, \mathbf{C})$ is:

$$\mathcal{H}(\mathcal{N}) = \log\left(\sqrt{(2\pi e)^n \det(\mathbf{C})}\right) , \tag{1}$$

where $\mathbf{C}$ is the covariance matrix. In order to obtain the most informative characterization of preferential search points, Kjellström designed GaA according to the following criteria: (a) The probability of finding an acceptable search point is fixed to a predefined value $P < 1$; (b) The spread of the samples, as quantified by their entropy, is to be maximized. As Eq. 1 shows, this can be achieved by maximizing the determinant of the covariance matrix. In order to minimize a real-valued objective function $f(\mathbf{x})$, GaA uses a fitness-dependent acceptance threshold c_T that is monotonically lowered until some convergence criteria are met.

1) The GaA algorithm: The GaA algorithm starts by setting the mean $\mathbf{m}^{(0)}$ of a multivariate Gaussian to an initial point $\mathbf{x}^{(\mathbf{0})} \in \mathcal{A}$. The covariance matrix $\mathbf{C}^{(g)}$ is decomposed as follows:

$$\mathbf{C}^{(g)} = \left(r \cdot \mathbf{Q}^{(g)}\right)\left(r \cdot \mathbf{Q}^{(g)}\right)^T = r^2 \left(\mathbf{Q}^{(g)}\right)\left(\mathbf{Q}^{(g)}\right)^T , \tag{2}$$

where r is the scalar *step size* of the algorithm and $\mathbf{Q}^{(g)}$ is the normalized square root of $\mathbf{C}^{(g)}$. Like in CMA-ES, $\mathbf{Q}^{(g)}$ is found by Cholesky or eigendecomposition of the covariance matrix $\mathbf{C}^{(g)}$. The initial $\mathbf{Q}^{(0)}$ is set to the identity matrix $\mathbf{I}$. The point at iteration $g+1$ is then sampled as:

$$\mathbf{x}^{(g+1)} = \mathbf{m}^{(g)} + r^{(g)}\mathbf{Q}^{(g)}\eta^{(g)} , \tag{3}$$

where $\eta^{(g)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The objective function is then evaluated at the position of the new sample, $f(\mathbf{x}^{(\mathbf{g+1})})$. *Only if* this fulfills $f(\mathbf{x}^{(g+1)}) < c_\mathrm{T}^{(g)}$, the following adaptation rules are applied: The step size r is increased according to:

$$r^{(g+1)} = f_\mathrm{e} \cdot r^{(g)} , \tag{4}$$

where $f_\mathrm{e} > 1$ is called the *expansion factor*. Increasing the step size after acceptance is a direct consequence of the maximum entropy principle. GaA tries to expand the distribution as much as possible for the given acceptance probability P, in order to maximize the determinant of the covariance matrix (see Eq. 1). Hence, r depends on P. The mean is updated as

$$\mathbf{m}^{(g+1)} = \left(1 - \frac{1}{N_\mathrm{m}}\right)\mathbf{m}^{(g)} + \frac{1}{N_\mathrm{m}}\mathbf{x}^{(g+1)} . \tag{5}$$

N_m is a weighting factor that controls the influence of the new sample on the mean. The covariance matrix is updated as

$$\mathbf{C}^{(g+1)} = \left(1 - \frac{1}{N_\mathrm{C}}\right)\mathbf{C}^{(g)} + \frac{1}{N_\mathrm{C}}(\Delta\mathbf{x})(\Delta\mathbf{x})^T , \tag{6}$$

where $\Delta\mathbf{x} = (\mathbf{x}^{(g+1)} - \mathbf{x}^{(g)})$. N_C weights the influence of the accepted sample point on the covariance adaptation. Kjellström also introduced an alternative update rule that is mathematically equivalent to Eq. 6, but numerically more robust. It acts directly on the square root $\mathbf{Q}^{(g)}$ of the covariance matrix:

$$\mathbf{\Delta C}^{(g+1)} = \left(1 - \frac{1}{N_\mathrm{C}}\right)\mathbf{I}^{(g)} + \frac{1}{N_\mathrm{C}}(\eta^{(g)})(\eta^{(g)})^T , \tag{7}$$

$$\mathbf{\Delta Q}^{(g+1)} = (\mathbf{\Delta C}^{(g+1)})^{\frac{1}{2}} . \tag{8}$$

$\mathbf{Q}^{(g+1)}$ is then computed as $\mathbf{Q}^{(g+1)} = \mathbf{Q}^{(g)}\mathbf{\Delta Q}^{(g+1)}$.

In order to decouple the volume of the covariance (controlled by $r^{(g+1)}$) and its orientation, $\mathbf{Q}^{(g+1)}$ is normalized such that $\det(\mathbf{Q}^{(g+1)}) = 1$. As in CMA-ES, the full adaptation of the covariance matrix gives GaA the appealing property of being invariant to rotations of the problem.

In case $\mathbf{x}^{(g+1)}$ is rejected, i.e. $f(\mathbf{x}^{(g+1)}) \geq c_\mathrm{T}^{(g)}$, the step size is reduced as:

$$r^{(g+1)} = f_\mathrm{c} \cdot r^{(g)} , \tag{9}$$

and neither the mean nor the covariance matrix are adapted. The *contraction factor* $f_\mathrm{c} < 1$ also depends on P.

In order to use GaA for optimization, the acceptance threshold c_T is continuously lowered. Kjellström proposed the following rule:

$$c_\mathrm{T}^{(g+1)} = \left(1 - \frac{1}{N_\mathrm{T}}\right) c_\mathrm{T}^{(g)} + \frac{1}{N_\mathrm{T}} f(\mathbf{x}^{(g+1)}) \,, \qquad (10)$$

where N_T controls the weighting between the old threshold and the objective value of the *accepted* sample. It can readily be seen that this fitness-dependent threshold update renders the algorithm invariant to linear transformations of the objective function.

2) Strategy parameters of GaA: The behavior of GaA is controlled by several strategy parameters. In the original publications, Kjellström investigated certain parameter settings in detail, while others have not been reported. We first consider the acceptance probability P. Kjellström analyzed the information-theoretic optimality of P for a random walk in a simplex region [9] and for GaA in general regions [2]. In both cases, he concluded that the efficiency E of the process and P are related as $E \propto -P \log P$, leading to an optimal $P = \frac{1}{e} \approx 0.3679$, where e is Euler's number. A proof is provided in Ref. [11]. Maintaining this optimal hitting probability corresponds to leaving the volume of the distribution, measured by $\det(\mathbf{C})$, constant under stationary conditions. Since $\det(\mathbf{C}) = r^{2n} \det(\mathbf{Q}\mathbf{Q}^T)$, the expansion and contraction factors f_e and f_c expand or contract the volume by a factor of f_e^{2n} and f_c^{2n}, respectively. After S accepted and F rejected samples, a necessary condition for constant volume thus is:

$$\prod_{i=1}^{S} (f_\mathrm{e})^{2n} \prod_{i=1}^{F} (f_\mathrm{c})^{2n} = 1 \,. \qquad (11)$$

Using $P = \frac{S}{S+F}$, and introducing a small $\beta > 0$, the choice $f_\mathrm{e} = 1 + \beta(1-P)$ and $f_\mathrm{c} = 1 - \beta P$ satisfies Eq. 11 to first order. The scalar rate β is coupled to N_C. We previously suggested the following rules of thumb for the parameter settings [7]: N_C influences the update of $\mathbf{C} \in \mathbb{R}^{n \times n}$, which contains n^2 entries. Hence, N_C should be related to n^2. We suggest using $N_\mathrm{C} = \frac{(n+1)^2}{\log(n+1)}$ as a standard value, and coupling $\beta = \frac{1}{N_\mathrm{C}}$ [7]. A similar reasoning is also applied to N_m. Since N_m influences the update of $\mathbf{m} \in \mathbb{R}^n$, it is reasonable to set $N_\mathrm{m} \propto n$. We propose $N_\mathrm{m} = en$ as a standard value. The setting for N_T will be addressed in Section II-B.3 below.

3) Relation between GaA and ES: There are several remarkable connections between GaA and classical ES. The canonical (1+1)-ES is, for example, a limit case of GaA. Setting $N_\mathrm{m} = N_\mathrm{T} = 1$ moves GaA's mean directly to the accepted sample and c_T to the fitness of the accepted sample. For $N_\mathrm{C} \to \infty$, the covariance remains isotropic and GaA becomes equivalent to the (1+1)-ES with a P^{th}-success rule. Keeping N_C finite results in an algorithm that is almost equivalent to the (1+1)-CMA-ES [12]. Four key differences, however, remain. First, the step size adaptation mechanism in (1+1)-CMA-ES uses a damped exponential function, allowing faster adaptation than in GaA [12]. Second, (1+1)-CMA-ES uses information about the evolution path for covariance matrix update, whereas GaA does not. Third, the decision of how to update the covariance is controlled by a threshold probability p_thresh in (1+1)-CMA-ES. Only if the empirical acceptance probability P_emp is below p_thresh, the current sample is used to update the evolution path. Finally, GaA normalizes the volume of the covariance matrix in order to decouple it from the step size; (1+1)-CMA-ES does not involve such a normalization.

B. Restart GaA

We introduce practical constraint handling and stopping criteria for GaA and extend it to Restart GaA, a quasi parameter-free black-box optimizer.

1) Constraint handling and initialization: In unconstrained optimization problems, GaA can be used as is. The starting point $\mathbf{m}^{(0)}$ and the initial step size $r^{(0)}$ then have to be set by the user.

In box-constrained optimization problems, boundaries are explicitly given by $\mathbf{x} \in [\mathbf{L}, \mathbf{U}] \subset \mathbb{R}^n$. Several boundary handling techniques can be employed. One can, e.g., reject points that fall outside the admissible hyper-rectangle, and resample. This can, however, become inefficient for search near the boundary. Especially in high dimensions, the probability of hitting the feasible region becomes small. It is also conceivable to employ boundary handling with quadratic penalty terms [13], a method that has been successfully used in CMA-ES. In GaA, however, the boundary penalty would be problem specific, since GaA's search performance directly depends on the objective function values. We therefore suggest projecting the components of out-of-bounds samples onto the boundary along the coordinate axes, and evaluating the projected samples.

The initial mean $\mathbf{m}^{(0)}$ is drawn from a uniform distribution in the box $[\mathbf{L}, \mathbf{U}]$. The initial step size is set to $r^{(0)} = 1/e \, (\max \mathbf{U} - \min \mathbf{L})$, similar to the global search setting of the initial σ in IPOP-CMA-ES [14]. The initial threshold $c_\mathrm{T}^{(0)}$ is set to $f(\mathbf{m}^{(0)})$.

2) Convergence criteria: In practical applications, it is often useful to define criteria that indicate convergence of GaA to a (local or global) minimum. We propose six convergence criteria: *MaxIter*, *TolFit*, *TolFun*, *TolX*, *TolR*, and *TolCon*.

1) *MaxIter*: GaA is stopped when a maximum number of allowed function evaluations is reached. The default maximum is $10^4 \, n$.
2) *TolFit*: If knowledge about the function value of the global minimum $f(\mathbf{x}_\mathrm{min})$ is available, the algorithm stops when the current best function value drops below $f(\mathbf{x}_\mathrm{min}) + \mathit{TolFit}$. The default setting is $\mathit{TolFit} = 10^{-9}$.
3) *TolFun*: If no knowledge about $f(\mathbf{x}_\mathrm{min})$ is available, GaA is considered converged when $\| \max f(\mathbf{x}^{(i)}) - \min f(\mathbf{x}^{(i)}) \| < \mathit{TolFun} \; \forall i \in [g-h; \, g]$. By default, we set $\mathit{TolFun} = 10^{-9}$ and the history length $h = 100$.

4) *TolX*: GaA is considered converged when $\|\mathbf{x}^{(g-h)} - \mathbf{x}^{(g)})\| < \mathit{TolX}$. By default, we set $\mathit{TolX} = 10^{-12}$ and the history length $h = 100$.
5) *TolR*: GaA is stopped when the step size $r^{(g)} < \mathit{TolR}$. The default setting is $\mathit{TolR} = 10^{-9}$.
6) *TolCon*: GaA is stopped when the difference between the current threshold $c_{\mathrm{T}}^{(g)}$ and the current best fitness value $f(\mathbf{x}_{\mathrm{best}}^{(g)})$ has converged, i.e., $\|f(\mathbf{x}_{\mathrm{best}}^{(g)}) - c_{\mathrm{T}}^{(g)})\| < \mathit{TolCon}$. The default setting is $\mathit{TolCon} = 10^{-9}$.

These stopping criteria are designed to reduce the number of non-improving function evaluations. They can directly be used to develop an effective restart strategy for GaA as outlined next.

3) Restart strategy: Depending on the topology of the optimization problem, canonical GaA with standard parameter settings may suffer from premature convergence to a suboptimal solution. This can be relaxed by introducing a restart mechanism that modifies the strategy parameters whenever any of the convergence criteria 3) to 6) above are met. In CMA-ES, the restart with iteratively increasing population size (IPOP-CMA-ES) proved powerful both on synthetic and real-world problems. Since GaA always samples a *single* candidate solution per iteration, the population size cannot be varied. Instead, we adapt the parameter N_{T} that controls the lowering of the fitness threshold c_{T}. The parameters N_{m}, N_{C}, β, and P are kept constant for all restarts. The initial value of N_{T} is $N_{\mathrm{T}}^{(0)} = N_{\mathrm{m}} = en$. At each restart i we *increase* $N_{\mathrm{T}}^{(i)}$ as

$$N_{\mathrm{T}}^{(i)} = r_{\mathrm{T}} N_{\mathrm{T}}^{(i-1)} \tag{12}$$

with $r_{\mathrm{T}} = 2$. The new initial starting point at each restart can either be chosen at random or at the converged position. The latter strategy is expected to be beneficial on funneled landscapes, such as Rastrigin's or Ackley's function.

The modification of $N_{\mathrm{T}}^{(i)}$ has a similar effect on GaA as increasing the population size has on CMA-ES. Initially, accepted samples are able to pull down the fitness threshold quickly. On unimodal functions, fast convergence is hence achieved. With increasing $N_{\mathrm{T}}^{(i)}$, c_{T} decreases slower and GaA has more time to explore the space and adapt a maximum-entropy distribution to the underlying problem structure.

C. Metropolis GaA

When using GaA for optimization, sample points that have a function value higher than c_{T} are strictly rejected and points with lower values accepted. We extend GaA to general adaptive sampling by replacing this hard threshold with an acceptance-rejection scheme [4]. We consider the case of black-box sampling, where the continuous target probability distribution $\pi(\mathbf{x})$ is only known up to a normalization constant, hence $f(\mathbf{x}) \propto \pi(\mathbf{x})$. This situation frequently occurs in Bayesian statistics and in statistical physics, where the unknown normalization constant is given by the partition function of the state space. In order to sample from $\pi(\mathbf{x})$, Metropolis introduced the following MCMC scheme: Define a symmetric proposal distribution $q(\cdot|\mathbf{x}^{(g)})$ that is easy to sample from. A standard choice is the multivariate isotropic Gaussian distribution $\mathcal{N}\big(\mathbf{x}^{(g)}, \sigma_n^2\mathbf{I}\big)$, where σ_n is the n-dependent scalar standard deviation. When a new candidate point $\mathbf{y}$ is sampled from $q(\cdot|\mathbf{x}^{(g)})$, it is accepted with probability

$$\alpha(\mathbf{x}^{(g)}, \mathbf{y}) = \min\left(1,\, \frac{f(\mathbf{y})}{f(\mathbf{x}^{(g)})}\right). \tag{13}$$

This means that if $f(\mathbf{y}) \geq f(\mathbf{x}^{(g)})$, it is always accepted; otherwise, it is accepted if $s \leq \alpha(\mathbf{x}^{(g)}, \mathbf{y})$ for s drawn from the standard uniform distribution. Upon acceptance, $\mathbf{x}^{(g+1)} = \mathbf{y}$ and the process continues with sampling from $\mathcal{N}\big(\mathbf{x}^{(g+1)}, \sigma_n^2\mathbf{I}\big)$. It can be shown that the set of accepted points represents an unbiased sample from the target distribution $\pi(\mathbf{x})$, i.e., that the Markov chain is *ergodic*.

The choice of the scalar standard deviation σ_n is, however, crucial for the efficiency of the sampling process. For n-dimensional Gaussian target distributions, the optimal choice is $\sigma_n = 2.4/\sqrt{n}$ [15].

The Metropolis algorithm constructs a Markov chain, where points at iteration (g) only depend on the previous point, i.e., the history of accepted points is discarded. This was extended by Haario and co-workers, who introduced the Adaptive Proposal (AP) algorithm [8] in which the proposal distribution depends on h previously accepted points, hence $q(\cdot|\mathbf{x}^{(g)}, \ldots, \mathbf{x}^{(g-h)})$. This history is used to adapt the covariance matrix of the Gaussian proposal. Although ergodicity has not been proven for this scheme, Haario et al. have empirically shown that AP improves the mixing of the chain considerably and hence yields superior performance on several target distributions.

Here, we introduce the Metropolis GaA (M-GaA) algorithm for continuous black-box sampling. We obtain M-GaA by replacing the acceptance rule $f(\mathbf{x}^{(g+1)}) < c_{\mathrm{T}}^{(g)}$ with the Metropolis criterion given in Eq. 13. We further set $N_{\mathrm{m}} = 1$, moving GaA's mean directly to the accepted sample $\mathbf{x}^{(g+1)}$. This yields a sampling algorithm with adaptive Gaussian proposals. A convenient feature of M-GaA is the possibility of setting the acceptance probability P *a priori*. This is not possible in the classical Metropolis algorithm, nor in the AP algorithm.

III. Benchmark Problems

We introduce the benchmark cases used to evaluate the performance of Restart GaA and M-GaA.

A. The CEC 2005 benchmark test suite

We test Restart GaA on the 25 benchmark functions defined during the CEC 2005 Special Session on Real-Parameter Optimization [16]. The CEC 2005 test suite provides a standard benchmark for real-valued optimization algorithms, along with standardized evaluation criteria and testing protocols. It thus allows comparing the performance of different optimization algorithms across publications. Functions f1 to f5 are unimodal and f6 to f12 are basic multimodal. Functions f13 and f14 are expanded, and functions f15 to f25 are hybrid test functions that are formed

by combining several elementary test functions. Functions f6, f11 to f13, and f15 to f25 are multi-funnel functions, where local optima cannot be interpreted as perturbations to a globally convex (unimodal) topology [17]. In order to prevent exploitation of search space symmetry, all functions are shifted and many of them are rotated. Moreover, the global optimum of each function is different from the usual zero value. Functions f4 and f17 are additionally corrupted by multiplicative white noise.

Following the evaluation criteria of the CEC 2005 test suite [16], we benchmark Restart GaA in $n = 10$, 30, and 50 dimensions. Each optimization run is repeated 25 times with uniformly random starting points inside the specified domain. The maximum allowed number of function evaluations (identical to *MaxIter* for GaA) is coupled to the problem dimension as: $\mathit{MaxIter} = 10^4\, n$.

We measure the performance of Restart GaA by the success rate $p_s = \#\text{successful runs}/\#\text{runs}$. A run is counted successful if the function error reaches a given accuracy *TolFit* (specified in the CEC 2005 benchmark) before *MaxIter* is reached.

B. Haario's twisted Gaussian distributions

In order to assess the performance of M-GaA as an adaptive sampler, we follow the protocol outlined in Ref. [8]. We consider the same three test target distributions:

π_1: Uncorrelated Gaussian distribution
π_2: Moderately twisted Gaussian distribution
π_3: Strongly twisted Gaussian distribution

Distribution π_1 is a centered n-dimensional multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{C_1})$ with $\mathbf{C_1} = diag(100, 1, \ldots, 1)$. It thus has the shape of an axis-aligned hyper-ellipsoid with an axes aspect ratio of 10. The twisted Gaussians are constructed as follows: Let g be the density of π_1. The density function of a twisted Gaussian with twisting parameter $b > 0$ is then given by

$$g_b = g(\Phi_b(\mathbf{x}))\,, \tag{14}$$

where $\Phi_b(\mathbf{x}) = (x_1, x_2 + bx_1^2 - 100b, x_3, \ldots, x_n)$. Φ_b thus only affects the second coordinate, and the determinant of its Jacobian is unity [8]. It is easy to compute probability regions of g_b and to verify that the expectation value of g_b is $\mathbf{0}$ for all b. Haario et al. used $b = 0.03$ for π_2 and $b = 0.1$ for π_3. Figure 1 shows the contour lines of the 68.3% and 99% probability regions of π_1 to π_3. Haario et al. also suggested the following quality measures for sampling algorithms:

1) mean($\|E\|$): The mean distance of the expectation values from their true value ($\mathbf{0}$), averaged over N repetitions
2) std($\|E\|$): The standard deviation of the distance of the expectation values from their true value, averaged over N repetitions
3) err($\leq 68.3\%$): The mean error (in %) of the percentage of sampled points that hit the probability region inside the 68.3% contour
4) std($\leq 68.3\%$): The standard deviation of err($\leq 68.3\%$)

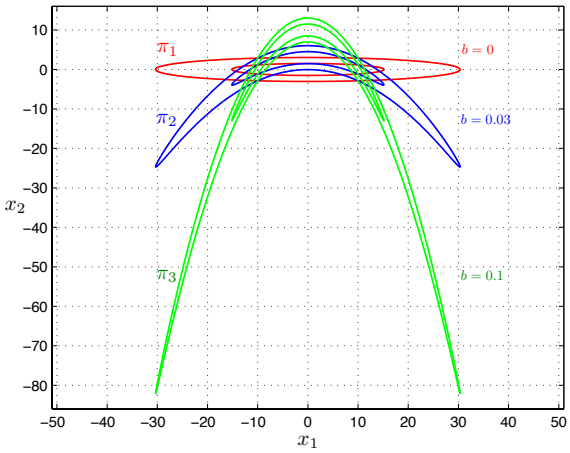

Fig. 1. 68.3% and 99% probability regions of the three test target distributions π_1 (red), π_2 (blue), and π_3 (green) in 2D. The parameter b controls the distortion of the Gaussian density (see main text for details).

5) err($> 99\%$): The mean error (in %) of the percentage of sampled points that hit the probability region outside the 99% contour.
6) std($> 99\%$): The standard deviation of err($> 99\%$)

IV. Benchmark Results

We summarize the benchmark results for Restart GaA and M-GaA on the test problems described above.

A. Restart GaA for black-box optimization

We evaluate the performance of Restart GaA on all 25 CEC 2005 test functions in $n = 10$, 30, and 50 dimensions. These results should be compared to those from IPOP-CMA-ES, the winning strategy on this benchmark in the 2005 competition, as tabulated in Ref. [14]. Table I summarizes the results for Restart GaA on the functions that can be solved within the predefined maximum number of function evaluations.

In $n = 10$ dimensions (Table I, upper panel), Restart GaA is able to solve f1 to f12, except for the needle-in-a-haystack problem f8. IPOP-CMA-ES is able to solve the identical set of functions (see Ref. [14] for results). Functions f1 to f7 are solved with $p_s = 1$. The pair f9/f10 (shifted/rotated Rastrigin) is solved with a lower success probability. Functions f11 (shifted Weierstrass) and f12 (Schwefel's problem), two multi-funnel functions, are solved with $p_s \geq 0.64$.

In $n = 30$, Restart GaA solves f1 to f7, except f5, with $p_s \geq 0.92$, as well as f11 with high and f12 with low probability. The Rastrigin pair f9/f10 can not be solved any more in 30 dimensions. Similar observations are made for $n = 50$. There, f1 to f4, f7, and f11 can be solved, but neither the Rastrigin pair nor f5/f12 are solved. Closer inspection of the results for f6 (shifted Rosenbrock) reveals that Restart GaA gets close to the minimum, but does not reach the specified accuracy in time.

The invariance of GaA to linear transformations of the search space is verified on the triple f1/f2/f3, the shifted

TABLE I

NUMBER OF FUNCTION EVALUATIONS (MIN, MEDIAN, MAXIMUM, MEAN, AND STANDARD DEVIATION) NEEDED BY RESTART GAA TO REACH $f(\mathbf{x}_{\text{MIN}}) + \textit{TolFit}$ FOR THE SOLVED FUNCTIONS IN $n = 10, 30, 50$ WITH LESS THAN 10^5 FUNCTION EVALUATIONS. THE LAST COLUMN SHOWS THE EMPIRICAL SUCCESS RATES p_s.

n=10						
Func.	**min**	**median**	**max**	**mean**	**std**	p_s
f1	7.65e+03	8.07e+03	8.33e+03	8.07e+03	1.88e+02	1
f2	7.80e+03	8.31e+03	8.56e+03	8.25e+03	2.05e+02	1
f3	1.09e+04	1.18e+04	1.56e+04	1.21e+04	1.17e+03	1
f4	7.77e+03	8.28e+03	1.89e+04	8.64e+03	2.14e+03	1
f5	7.28e+03	8.20e+03	-	1.63e+04	1.83e+04	0.96
f6	1.82e+04	2.04e+04	2.53e+04	2.08e+04	1.86e+03	1
f7	5.11e+03	5.46e+03	5.90e+03	5.45e+03	1.91e+02	1
f8	-	-	-	-	-	-
f9	3.74e+04	-	-	5.72e+04	2.81e+04	0.08
f10	3.92e+03	-	-	4.01e+04	3.70e+04	0.12
f11	1.37e+04	4.08e+04	-	4.36e+04	2.22e+04	0.80
f12	5.64e+03	3.10e+04	-	2.61e+04	1.77e+04	0.64
n=30						
Func.	**min**	**median**	**max**	**mean**	**std**	p_s
f1	4.34e+04	4.43e+04	4.51e+04	4.42e+04	4.11e+02	1
f2	4.56e+04	4.67e+04	4.76e+04	4.67e+04	4.81e+02	1
f3	7.19e+04	7.93e+04	8.97e+04	7.91e+04	4.44e+03	1
f4	9.63e+04	1.01e+05	2.00e+05	1.05e+05	1.99e+04	1
f5	-	-	-	-	-	-
f6	1.42e+05	2.51e+05	-	2.47e+05	3.63e+04	0.92
f7	2.98e+04	3.05e+04	3.17e+04	3.06e+04	4.73e+02	1
f8	-	-	-	-	-	-
f9	-	-	-	-	-	-
f10	-	-	-	-	-	-
f11	8.42e+04	2.70e+05	-	2.25e+05	6.04e+04	0.80
f12	1.75e+05	-	-	1.75e+05	-	0.04
n=50						
Func.	**min**	**median**	**max**	**mean**	**std**	p_s
f1	9.82e+04	9.95e+04	1.00e+05	9.94e+04	5.53e+02	1
f2	1.06e+05	1.09e+05	1.11e+05	1.09e+05	1.48e+03	1
f3	1.94e+05	2.04e+05	2.20e+05	2.06e+05	6.97e+03	1
f4	2.11e+05	2.18e+05	4.32e+05	2.64e+05	8.79e+04	1
f5	-	-	-	-	-	-
f6	-	-	-	-	-	-
f7	6.63e+04	6.77e+04	6.87e+04	6.77e+04	5.52e+02	1
f8	-	-	-	-	-	-
f9	-	-	-	-	-	-
f10	-	-	-	-	-	-
f11	3.55e+05	-	-	4.49e+05	5.87e+04	0.36
f12	-	-	-	-	-	-

sphere, Schwefel's problem, and the high-conditional ellipsoid. In $n = 10$ and 30, the mean number of function evaluations used is almost identical for f1 and f2. For f3, GaA needs more samples for properly adapting its covariance matrix. These results are similar to those of IPOP-CMA-ES [14]. The performance of IPOP-CMA-ES, however, scales better with the dimensionality n, especially for the sphere [14]. As opposed to IPOP-CMA-ES, Restart GaA cannot solve problem f5 for $n \geq 30$ where the global minimum is located at the bounds

The failure of Restart GaA on the Rastrigin pair f9/f10 in $n = 30$ and 50 indicates that population-based methods outperform single-sample strategies on landscapes with many local minima near the global one. Restart GaA, however, shows good robustness against noise. Function f4 is the same as f2, but with multiplicative white noise. While IPOP-CMA-ES cannot solve this function in $n = 50$, Restart GaA solves it without problems. The noise does not hamper the maximum-entropy adaptation in GaA. It is also noteworthy

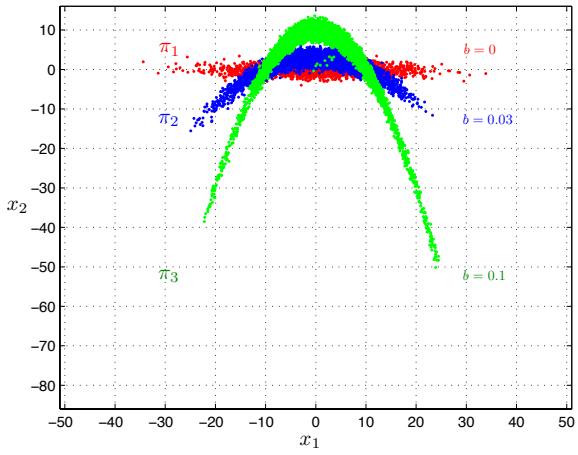

Fig. 2. Complete set of M-GaA samples from the test target distributions π_1 (red), π_2 (blue), and π_3 (green) for one run, randomly selected from the 100 runs. A 2D projection of the 8-dimensional data set is shown.

that the multi-funnel function f11 can be efficiently solved by Restart GaA in all tested dimensions. IPOP-CMA-ES solves f11 only in 10 and 30 dimensions, albeit with a much lower success rate.

In order to test the efficacy of the proposed new restart procedure, we repeat all tests with $r_{\text{T}} = 1$, i.e., without adapting N_{T} upon restart (data not shown). For f1 to f3, restart was never needed. In all other cases, doubling N_{T} upon restart leads to superior performance in *all* problem instances.

B. M-GaA for adaptive sampling

We test the M-GaA sampling scheme on the target distributions π_1 to π_3 in $n = 8$ dimensions [8]. 100 independent runs are performed for each target. The sample size is limited to $20,000$ for π_1, $40,000$ for π_2, and $80,000$ for π_3. In all test cases, M-GaA's initial sample point is drawn uniformly at random within the hyper-cube $[-1, 1]^8$, and the initial step size is $r^{(0)} = 1$. Since the chain in M-GaA rapidly mixes, the burn-in length is set to $1,000$. In order to be close the empirical acceptance probability of the AP algorithm (0.14 for π_2 and 0.09 for π_3 [8]), the hitting probability P is set to 0.1 in all cases. Figure 2 shows 2D projections of some M-GaA samples from each target distribution.

We compare the performance of M-GaA to three other algorithms:

1) Single-component Metropolis algorithm (SC) with univariate Gaussian proposal. This algorithm explores each coordinate axis separately, one after the other [4].
2) Metropolis-Hastings algorithm (MH) with isotropic multivariate Gaussian proposal. This algorithm explores all directions simultaneously.
3) Adaptive Proposal Random Walk Monte Carlo (AP): This algorithm adapts the covariance of the multivariate Gaussian proposal based on a finite history of accepted samples.

For both SC and MH, the standard deviation of the Gaussian proposal is fixed to the optimal value of $2.4/\sqrt{n}$. For AP we

TABLE II

SUMMARY STATISTICS OF 100 INDEPENDENT TEST RUNS OF SINGLE-COMPONENT METROPOLIS (SC), METROPOLIS-HASTINGS (MH), ADAPTIVE PROPOSAL (AP) (TAKEN FROM [8]), AND M-GAA SAMPLERS. ALL **ERR** AND **STD** VALUES ARE GIVEN IN %.

π_1				
	SC	**MH**	**AP**	**M-GaA**
mean($\|E\|$)	-	2.96	0.46	0.62
std($\|E\|$)	-	2.31	0.33	0.44
err($\leq 68.3\%$)	-	0.23	0.02	4.29
std($\leq 68.3\%$)	-	4.40	1.95	2.41
err($> 99\%$)	-	0.01	0.03	0.04
std($> 99\%$)	-	0.61	1.32	0.39
π_2				
	SC	**MH**	**AP**	**M-GaA**
mean($\|E\|$)	2.40	2.46	1.31	1.48
std($\|E\|$)	4.59	2.81	0.72	0.71
err($\leq 68.3\%$)	1.30	0.18	0.80	0.29
std($\leq 68.3\%$)	4.59	6.70	2.92	1.95
err($> 99\%$)	0.16	0.03	0.01	0.16
std($> 99\%$)	0.40	0.66	0.62	0.25
π_3				
	SC	**MH**	**AP**	**M-GaA**
mean($\|E\|$)	6.53	7.89	4.85	4.96
std($\|E\|$)	4.79	7.54	4.20	1.14
err($\leq 68.3\%$)	2.46	0.35	2.13	1.27
std($\leq 68.3\%$)	6.48	9.79	5.34	2.56
err($> 99\%$)	0.27	0.07	0.14	0.26
std($> 99\%$)	0.34	0.97	0.45	0.28

use the parameter values given in Ref. [8]. The burn-in length is set to 50% of the sample size for all three algorithms [8].

The performance measures for all algorithms are summarized in Table II. All results other than those for M-GaA are taken from Ref. [8]. For the uncorrelated target π_1, M-GaA and AP both outperform MH in estimating the expectation value. They also show a lower standard deviation of the estimation. While the samples from M-GaA have a bias of around 4% in the 68.3% region, they accurately cover the tails beyond the 99% region. For the twisted Gaussians π_2 and π_3, M-GaA and AP estimate the expectation more accurately than SC and MH. This indicates that M-GaA is able to better explore the twisted tails of the distributions, leading to a smaller error in the expectation estimation. For all twisted distributions, the M-GaA estimates have smaller standard deviations than those from any of the other algorithms.

We study the mixing behavior of the algorithms by computing the component-wise autocorrelation R of the M-GaA and MH samples (Figure 3). For π_1, the sample components x_1 (along the stretched axis) are much less correlated in M-GaA than they are in MH. The same is true for π_2 in both the first and second dimension (stretched and twisted). All other components show low correlations in both algorithms, with the MH sample autocorrelation dropping slightly faster than that of M-GaA (curves in the lower-left corner of the graphs in Fig. 3). We observe the same behavior also for the target π_3 (data not shown). Altogether, the strong reduction in sample autocorrelation indicates fast mixing of the chains produced by M-GaA.

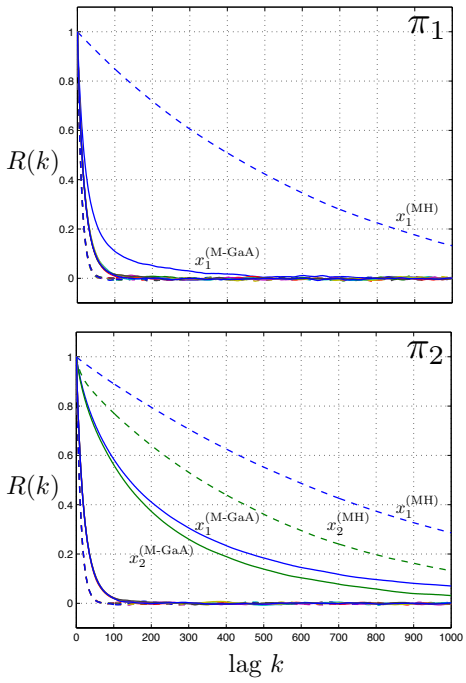

Fig. 3. Component-wise autocorrelation R of the samples (after the burn-in phase) vs. lag k on π_1 and π_2 for MH (dashed lines) and M-GaA (solid lines) averaged over 100 runs. The M-GaA samples in the non-standard-normal coordinates (x_1 in π_1 and x_1, x_2 in π_2) are less correlated than the corresponding components in the MH samples. In all standard-normal components, the sample autocorrelation drops fast for both algorithms (≈ 0 at $k = 150$).

V. DISCUSSION

Exploring the common concepts in MCMC samplers and ES, we presented a unifying framework for continuous black-box optimization and adaptive sampling based on GaA. We have presented empirical evidence for the efficiency and efficacy of the GaA-based algorithms in comparison to the respective state-of-the-art black-box optimization and adaptive sampling schemes.

We tested the newly introduced Restart GaA on the complete CEC 2005 test suite. These benchmarks verified the expected invariance of GaA to linear transformations of the search space. Moreover, Restart GaA proved to be robust against noise, as indicated by its performance on the noisy, shifted Schwefel function f4 in 50 dimensions. Restart GaA also showed a remarkable ability to solve the fractal multi-funnel Weierstrass function f11 in all tested dimensions. A clear limitation of Restart GaA is its inability to solve the Rastrigin functions f9/f10 in $n = 30$ and 50. Adaptation of the threshold c_T is too fast, resulting in premature convergence in a local minimum close to the global one. The benchmarks suggest that population-based CMA methods (such as IPOP-CMA-ES) should be preferred over algorithms with single candidate samples for functions with many local minima superimposed on a globally unimodal structure (as f9/f10). Restart GaA also suffers from reduced performance when the global optimum is located at the boundary of the

search space in higher dimensions (function f5). This could potentially be addressed by a more sophisticated boundary handling mechanism than the one proposed here.

We benchmarked the sampling performance of M-GaA using the protocol given in Ref. [8]. In all cases, M-GaA proved to be competitive with MH and AP samplers. It accurately estimated the first moment and correctly covered all probability regions, even in the case of highly distorted targets. M-GaA's hitting probability P has been set to 0.1 in order to be comparable with the AP algorithm. According to Gelman et al. [15], the optimal acceptance rate of the MH algorithm for Gaussian targets is around 0.234. We thus expect to be able to further improve M-GaA's performance by tuning its acceptance probability. AP and M-GaA do not produce a Markov chain because the proposal at each step depends on several previously accepted points. Hence, it has not been proven that these algorithms draw unbiased samples from any target distribution. M-GaA (like MH and AP) is also expected to experience problems when sampling from multi-modal target distributions where the modes are separated by large regions of low probability.

VI. Conclusions and Outlook

We have extended GaA, a stochastic optimization method that dates back to the 1960's, by two mechanisms: First, we introduced general parameter settings, convergence criteria, and a restart mechanism for continuous black-box optimization. The resulting Restart GaA is a *quasi* parameter-free black-box optimizer. Second, we introduced M-GaA, where sample acceptance is controlled by a Metropolis-like acceptance-rejection criterion. This led to a sampling scheme with adaptive Gaussian proposals.

We empirically tested both extensions on well-known benchmark problems. The performance of Restart GaA has been evaluated on the CEC 2005 benchmark test suite. The results indicate that the algorithm is a competitive choice for noisy and multi-funnel optimization problems. M-GaA outperformed standard Metropolis algorithms on distorted unimodal target distributions, and proved to be competitive with the AP algorithm. The concept of Gaussian Adaptation, and its theoretical roots in maximum-entropy sampling, thus present a unifying framework for both black-box optimization and Monte Carlo sampling.

Several practical improvements to the present framework are conceivable. For Restart GaA, effective penalty terms for constraint handling could be designed. When facing multi-modal target distribution, M-GaA could also be augmented with a restart mechanism in order to explore several modes sequentially. Parallel M-GaA runs that periodically exchange proposal samples might also be considered for this purpose.

We plan to test Restart GaA on real-world problems, especially on robust parameter estimation for biological and electrical networks.

Future theoretical work will be guided by recent results on adaptive MCMC methods [18], [19]. We will specifically address the question under which conditions a proof of ergodicity for M-GaA is feasible.

Acknowledgments

We thank Janick Cardinale and Dr. Grégory Paul from our group for many valuable and inspiring discussions.

References

[1] H. Haario and E. Saksman, "Simulated Annealing in General State-Spaces," *Advances in Applied Probability*, vol. 23, no. 4, pp. 866–893, DEC 1991.

[2] G. Kjellström and L. Taxen, "Stochastic Optimization in System Design," *IEEE Trans. Circ. and Syst.*, vol. 28, no. 7, July 1981.

[3] P. Green and X. Han, "Metropolis methods, Gaussian proposals and antithetic variables," in *Stochastic Models, Statistical Methods and Algorithms in Image Analysis.*, P. Barone, A. Frigessi, and M. Piccioni, Eds. Berlin, Germany: Springer-Verlag, 1992, pp. 142–64.

[4] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[5] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer, October 2002.

[6] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaption in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[7] C. L. Müller and I. F. Sbalzarini, "Gaussian adaptation revisited - an entropic view on covariance matrix adaptation," in *EvoApplications*, ser. Lecture Notes in Computer Science, C. Di Chio et al., Ed., vol. I, no. 6024. Springer, 2010, pp. 432–441.

[8] H. Haario, E. Saksman, and J. Tamminen, "Adaptive proposal distribution for random walk Metropolis algorithm," *Computational Statistics*, vol. 14, no. 3, pp. 375–395, 1999.

[9] G. Kjellström, "Network Optimization by Random Variation of Component Values," *Ericsson Technics*, vol. 25, no. 3, pp. 133–151, 1969.

[10] E. T. Jaynes, "Information Theory and Statistical Mechanics," *Phys. Rev.*, vol. 106, no. 4, pp. 620–630, May 1957.

[11] G. Kjellström, "On the Efficiency of Gaussian Adaptation," *J. Optim. Theory Appl.*, vol. 71, no. 3, December 1991.

[12] C. Igel, T. Suttorp, and N. Hansen, "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*. New York, NY, USA: ACM, 2006, pp. 453–460.

[13] N. Hansen and S. Kern, "Evaluating the CMA Evolution Strategy on Multimodal Test Functions," in *Lecture Notes in Computer Science*, ser. Parallel Problem Solving from Nature – PPSN VIII. Springer, 2004, pp. 282–291.

[14] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size." in *Proc. of IEEE Congress on Evolutionary Computation (CEC 2005)*, vol. 2, 2005, pp. 1769–1776.

[15] A. Gelman, G. Roberts, and W. Gilks, "Efficient Metropolis jumping rules," in *Bayesian Statistics*, J. M. Bernado *et al.*, Eds. OUP, 1996, vol. 5, p. 599.

[16] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," Nanyang Technological University, Singapore, Tech. Rep., May 2005.

[17] C. L. Müller, B. Baumgartner, and I. F. Sbalzarini, "Particle Swarm CMA Evolution Strategy for the optimization of multi-funnel landscapes," in *Proc. of IEEE Congress on Evolutionary Computation (CEC 2009)*, May 2009, pp. 2685–2692.

[18] C. Andrieu and E. Moulines, "On the ergodicity properties of some adaptive MCMC algorithms," *Annals of Applied Probability*, vol. 16, no. 3, pp. 1462–1505, Aug. 2006.

[19] C. Andrieu and J. Thoms, "A tutorial on adaptive MCMC," *Statistics and Computing*, vol. 18, no. 4, pp. 343–373, Dec. 2008.