Active Covariance Matrix Adaptation for the (1+1)-CMA-ES

Dirk V. Arnold, Nikolaus Hansen

Active Covariance Matrix Adaptation for the (1+1)-CMA-ES

Dirk V. Arnold
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia
Canada B3H 1W5
dirk@cs.dal.ca

Nikolaus Hansen
INRIA Saclay – Île-de-France
Machine Learning and Optimization Group (TAO)
Université Paris-Sud, LRI, Bât. 490
91405 Orsay Cedex, France
Nikolaus.Hansen@lri.fr

ABSTRACT

We propose a novel variant of the $(1 + 1)$-CMA-ES that updates the distribution of mutation vectors based on both successful and unsuccessful trial steps. The computational costs of the adaptation procedure are quadratic in the dimensionality of the problem, and the algorithm retains all invariance properties. Its performance on a set of standard test functions is compared with that of the original strategy that updates the distribution of mutation vectors in response to successful steps only. The new variant is not observed to be more than marginally slower on any function, and it is up to two times faster on some of the test problems.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Unconstrained Optimization; I.2.8 [**Problem Solving, Control Methods, and Search**]; I.2.6 [**Learning**]: Parameter Learning

General Terms

Algorithms, Performance

Keywords

Stochastic optimisation, variable metric algorithm, evolution strategy, covariance matrix adaptation

1. INTRODUCTION

On ill-conditioned problems covariance matrix adaptation can accelerate the rate of convergence of evolution strategies by orders of magnitude. For example, successful covariance matrix adaptation can enable strategies to generate candidate solutions predominantly in the direction of narrow valleys. The covariance matrix adaptation evolution strategy (CMA-ES) developed by Hansen and Ostermeier [6] learns an appropriate covariance matrix from successful steps that the algorithm has taken. It exhibits desirable invariance properties that make it suitable for solving non-separable optimisation problems. Restart variants of the CMA-ES, such as that by Auger and Hansen [3], have dominated other algorithms in benchmarking exercises at the 2005 *IEEE Congress on Evolutionary Computation (CEC)*[1] as well as at the 2009 *Genetic and Evolutionary Computation Conference (GECCO)*[2].

The $(1 + 1)$-CMA-ES is a recent variant of the CMA-ES that has been introduced by Igel et al. [7] and developed further by Suttorp et al. [11]. It differs from previous CMA-ES variants in being elitist, and Igel et al. [7] find that it is about 1.5 times faster than the $(\mu/\mu, \lambda)$-CMA-ES on unimodal functions[3]. A particularly interesting feature of the (1+1)-CMA-ES is that it operates on the Cholesky factors of the covariance matrix, removing the need for computationally expensive eigenvalue or Cholesky decompositions. As a result, it is easy to implement and potentially particularly useful in scenarios where the costs of fitness evaluations are such that covariance matrix decompositions would dominate the computational costs of the algorithm.

The CMA-ES adapts the covariance matrix of the distribution of mutation vectors based on successful steps. The underlying idea is to increase the variance of the distribution in directions that have proven successful in the recent past. Old information present in the covariance matrix decays passively. Jastrebski and Arnold [8] propose what they refer to as active covariance matrix adaptation for the $(\mu/\mu, \lambda)$-ES. Active covariance matrix adaptation increases variances in successful directions as well as actively reducing variances in particularly unsuccessful ones. In experiments on standard benchmark functions they find that the strategy that employs active covariance matrix adaptation consistently outperforms the standard $(\mu/\mu, \lambda)$-CMA-ES. The advantage of the strategy that employs active adaptation is particularly pronounced on functions with an eigenvalue spectrum of the Hessian that is dominated by a small number of relatively large values.

In this paper we introduce active covariance matrix adaptation for the $(1 + 1)$-CMA-ES. The algorithm maintains all invariance properties of the CMA-ES, and its computational cost per time step (excluding fitness evaluations) is quadratic in the dimensionality of the search space. The remainder of the paper is organised as follows: Section 2 outlines the $(1 + 1)$-CMA-ES as described in [11]. Section 3 introduces active covariance matrix adaptation for

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

[1] `http://www3.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm`
[2] `http://coco.gforge.inria.fr/doku.php?id=bbob-2009`
[3] See Beyer and Schwefel [5] for an introduction to evolution strategy nomenclature.

the $(1+1)$-CMA-ES. Section 4 compares the performances of the algorithms with and without active covariance matrix adaptation on sets of standard test functions. Section 5 concludes with a discussion of the results and of future work.

2. (1+1)-CMA-ES

Applied to a minimisation problem with objective function $f : \mathbb{R}^n \to \mathbb{R}$, CMA-ES sample offspring candidate solutions from a distribution with covariance matrix $\sigma^2\mathbf{C}$, where $\mathbf{C} \in \mathbb{R}^{n\times n}$ and $\sigma \in \mathbb{R}$. Matrix $\mathbf{C}$ is usually referred to as the covariance matrix, scalar σ as the global step size. The covariance matrix is adapted by means of updates of the form

$$\mathbf{C} \leftarrow (1 - c^+_{\text{cov}})\mathbf{C} + c^+_{\text{cov}}\mathbf{s}\mathbf{s}^{\mathrm{T}} \tag{1}$$

where '$\leftarrow$' denotes the assignment operator, $\mathbf{s} \in \mathbb{R}^n$ is an exponentially fading record of recent steps referred to as the evolution or search path, and $c^+_{\text{cov}} > 0$ is a constant that determines the time scale on which old information present in the covariance matrix fades out. The global step size is adapted separately. The $(1+1)$-CMA-ES in particular employs a variant of Rechenberg's 1/5th-rule for that purpose.

In order to generate samples from a normal distribution with covariance matrix $\mathbf{C}$, a Cholesky decomposition $\mathbf{C} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$ is computed and mutation vectors are generated as $\mathbf{A}\mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^n$ consists of n independent, standard normally distributed components. The computational cost of the matrix decomposition is in $\Theta(n^3)$ and thus significant for high-dimensional problems. Hansen and Ostermeier [6] suggest decomposing the covariance matrix only every n/k time steps for some constant k, and to use outdated Cholesky factors in between. Ros and Hansen [10] propose a variant of the CMA-ES that constrains matrix $\mathbf{C}$ to be diagonal, reducing the task of computing a Cholesky decomposition to that of taking the square roots of the diagonal elements, but resulting in a loss of invariance with regard to rotations of the coordinate system.

The $(1+1)$-CMA-ES avoids computationally expensive matrix decompositions altogether. Instead of operating on the covariance matrix $\mathbf{C}$, it operates directly on the Cholesky factor $\mathbf{A}$ and its inverse $\mathbf{A}_{\text{inv}} = \mathbf{A}^{-1}$. Specifically, Suttorp et al. [11] let $\mathbf{w} = \mathbf{A}_{\text{inv}}\mathbf{s}$ and define

$$a = \sqrt{1 - c^+_{\text{cov}}} \tag{2}$$

and

$$b = \frac{\sqrt{1 - c^+_{\text{cov}}}}{\|\mathbf{w}\|^2}\left(\sqrt{1 + \frac{c^+_{\text{cov}}}{1 - c^+_{\text{cov}}}\|\mathbf{w}\|^2} - 1\right). \tag{3}$$

They then show that if $\mathbf{C} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$ and $\mathbf{A}_{\text{inv}} = \mathbf{A}^{-1}$, updates

$$\mathbf{A} \leftarrow a\mathbf{A} + b\,[\mathbf{A}\mathbf{w}]\,\mathbf{w}^{\mathrm{T}} \tag{4}$$

and

$$\mathbf{A}_{\text{inv}} \leftarrow \frac{1}{a}\mathbf{A}_{\text{inv}} - \frac{b}{a^2 + ab\|\mathbf{w}\|^2}\mathbf{w}\left[\mathbf{w}^{\mathrm{T}}\mathbf{A}_{\text{inv}}\right] \tag{5}$$

maintain those relationships while performing the covariance matrix update in Eq. (1). Notice that with the order of the computations performed as suggested by the square brackets, the update requires $\Theta(n^2)$ time.

The state of the $(1+1)$-CMA-ES consists of parental candidate solution $\mathbf{x} \in \mathbb{R}^n$, search path $\mathbf{s} \in \mathbb{R}^n$, global step size $\sigma \in \mathbb{R}^+$, success probability estimate $P_{\text{succ}} \in \mathbb{R}$, and Cholesky factor $\mathbf{A} \in \mathbb{R}^{n\times n}$ and its inverse $\mathbf{A}_{\text{inv}}$. In every iteration, those quantities are updated in three steps:

1. Generate offspring candidate solution $\mathbf{y} = \mathbf{x} + \sigma\mathbf{A}\mathbf{z}$, where $\mathbf{z}$ is a vector with n independent, standard normally distributed components.
2. If $f(\mathbf{y}) \le f(\mathbf{x})$, then do the following:
   (a) Let
   $$\mathbf{x} \leftarrow \mathbf{y}.$$
   (b) Update the success probability estimate according to
   $$P_{\text{succ}} \leftarrow (1 - c_P)P_{\text{succ}} + c_P$$
   where c_P is a constant with $0 < c_P < 1$.
   (c) Update the search path according to
   $$\mathbf{s} \leftarrow (1 - c)\mathbf{s} + \sqrt{c(2 - c)}\mathbf{A}\mathbf{z}.$$
   (d) Let $\mathbf{w} = \mathbf{A}_{\text{inv}}\mathbf{s}$ and update $\mathbf{A}$ and $\mathbf{A}_{\text{inv}}$ according to Eqs. (4) and (5) with the coefficients computed using Eqs. (2) and (3).

   Otherwise, update the success probability estimate according to
   $$P_{\text{succ}} \leftarrow (1 - c_P)P_{\text{succ}}$$
3. Update the global step size according to
   $$\sigma \leftarrow \sigma \exp\left(\frac{1}{d}\frac{P_{\text{succ}} - P_{\text{target}}}{1 - P_{\text{target}}}\right)$$
   where $d > 0$ is a damping constant and target success probability P_{target} equals approximately one fifth.

Settings for all constants can be found in [11].[4]

3. ACTIVE COVARIANCE MATRIX ADAPTATION

Equation (1) describes a rank-one update to the covariance matrix that increases variances in directions of previously successful steps accumulated in the search path. Active covariance matrix adaptation additionally performs a rank-one update of the form

$$\mathbf{C} \leftarrow (1 + c^-_{\text{cov}})\mathbf{C} - c^-_{\text{cov}}(\mathbf{A}\mathbf{z})(\mathbf{A}\mathbf{z})^{\mathrm{T}} \tag{6}$$

for constant $c^-_{\text{cov}} > 0$ if a candidate solution $\mathbf{y} = \mathbf{x} + \sigma\mathbf{A}\mathbf{z}$ is particularly unsuccessful. Like the update in Eq. (1), the

[4]The algorithm described in [11] differs from the one given here in that it stalls the update of the search path if the success probability estimate P_{succ} exceeds a threshold $P_{\text{thresh}} = 0.44$. Specifically, in that case, the update in 2.(c) is replaced with $\mathbf{s} \leftarrow (1-c)\mathbf{s}$. Additionally, in order to compensate for the influence of the shortened search path, the coefficients in Eqs. (2) and (3) are replaced with $a = \sqrt{1-d}$ and $b = \sqrt{1-d}(\sqrt{1 + c^+_{\text{cov}}\|\mathbf{w}\|^2/(1-d)} - 1)/\|\mathbf{w}\|^2$, respectively, where $d = c^+_{\text{cov}}(1 + c(2-c))$. While that feature is useful in situations where the global step size is much smaller than optimal, its influence on the performance of the algorithm is minor in most cases. We have omitted it from the description of the algorithm for clarity, but we have implemented it for the experimental evaluation in Section 4.

update in Eq. (6) is unbiased in that the expected value of the covariance matrix remains unchanged under random selection (i.e., if $\mathbf{z}$ is standard normally distributed). The update has the effect of reducing the variance of the distribution of mutation vectors in the direction of particularly unsuccessful steps.

In the case of the $(\mu/\mu, \lambda)$-ES considered in [8] "especially unsuccessful" is defined as being among the worst of the offspring generated in the current time step. For the $(1+1)$-ES the offspring population is of size one and an alternative definition is needed. We store the objective function values of recent parental candidate solutions and consider candidate solution $\mathbf{y}$ to be "especially unsuccessful" if its objective function value is inferior to that of its kth-order ancestor (where the first-order ancestor is the parent, the second-order ancestor the grandparent, etc.). Specifically, we use $k = 5$. That value can empirically be observed to often result in the probability of a candidate solution being labelled especially unsuccessful being approximately one fifth. As the relationship between that probability and k is monotonic, adaptive schemes for k are easily conceivable if desired.

It remains to provide an update of the Cholesky factor $\mathbf{A}$ and its inverse that implicitly results in the covariance matrix update in Eq. (6). Such an update can be derived immediately using the following theorem:

THEOREM 1. *Let $\mathbf{C} \in \mathbb{R}^{n\times n}$ be a symmetric, positive definite matrix with Cholesky factorisation $\mathbf{C} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$, and let*

$$\mathbf{C}' = \alpha\mathbf{C} + \beta\mathbf{v}\mathbf{v}^{\mathrm{T}}$$

where $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v} \neq \mathbf{0}$, $\alpha \in \mathbb{R}^+$, and $\beta \in \mathbb{R}$. Letting $\mathbf{w} = \mathbf{A}^{-1}\mathbf{v}$, provided that $\alpha + \beta\|\mathbf{w}\|^2 > 0$, matrix $\mathbf{C}'$ can be written as $\mathbf{C}' = \mathbf{A}'\mathbf{A}'^{\mathrm{T}}$ with

$$\mathbf{A}' = \sqrt{\alpha}\mathbf{A} + \frac{\sqrt{\alpha}}{\|\mathbf{w}\|^2}\left(\sqrt{1+\frac{\beta}{\alpha}\|\mathbf{w}\|^2} - 1\right)\mathbf{A}\mathbf{w}\mathbf{w}^{\mathrm{T}}.$$

Furthermore,

$$\mathbf{A}'^{-1} = \frac{1}{\sqrt{\alpha}}\mathbf{A}^{-1} - \frac{1}{\sqrt{\alpha}\|\mathbf{w}\|^2}\left(1 - \frac{1}{\sqrt{1+\beta\|\mathbf{w}\|^2/\alpha}}\right)\mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{A}^{-1}$$

holds in that case.

PROOF. For $\beta > 0$ the theorem is proven in [11]. The proof relies on the identity

$$\mathbf{I} + \mathbf{u}\mathbf{u}^{\mathrm{T}} = \left(\mathbf{I} + \varsigma\mathbf{u}\mathbf{u}^{\mathrm{T}}\right)\left(\mathbf{I} + \varsigma\mathbf{u}\mathbf{u}^{\mathrm{T}}\right)$$

where $\mathbf{I}$ is the $n \times n$ identity matrix and $\varsigma = (\sqrt{1+\|\mathbf{u}\|^2} - 1)/\|\mathbf{u}\|^2$ for $\mathbf{u} \neq \mathbf{0}$ and $\varsigma = 0$ otherwise. It is easily verified (by multiplying out the right hand side) that for $0 < \|\mathbf{u}\| \leq 1$ the identity

$$\mathbf{I} - \mathbf{u}\mathbf{u}^{\mathrm{T}} = \left(\mathbf{I} + \varsigma\mathbf{u}\mathbf{u}^{\mathrm{T}}\right)\left(\mathbf{I} + \varsigma\mathbf{u}\mathbf{u}^{\mathrm{T}}\right)$$

holds, where $\varsigma = (\sqrt{1-\|\mathbf{u}\|^2} - 1)/\|\mathbf{u}\|^2$. With this, showing that Theorem 1 holds for $\beta < 0$ is entirely analogous to the proof of Theorems 1 and 2 in [11]. That the theorem holds for $\beta = 0$ is obvious. □

Table 1: Parameter settings.

$d = 1 + \frac{n}{2}$	$c = \frac{2}{n+2}$	$c_P = \frac{1}{12}$	$P_{\text{target}} = \frac{2}{11}$
$c^+_{\text{cov}} = \frac{2}{n^2+6}$		$c^-_{\text{cov}} = \frac{0.4}{n^{1.6}+1}$	

Thus, from Theorem 1, the covariance matrix update in Eq. (6) can be accomplished implicitly by letting

$$a = \sqrt{1 + c^-_{\text{cov}}} \tag{7}$$

and

$$b = \frac{\sqrt{1+c^-_{\text{cov}}}}{\|\mathbf{z}\|^2}\left(\sqrt{1 - \frac{c^-_{\text{cov}}}{1+c^-_{\text{cov}}}\|\mathbf{z}\|^2} - 1\right) \tag{8}$$

and using Eqs. (4) and (5) to update the Cholesky factor and its inverse. The update is admissible only if

$$1 - \frac{c^-_{\text{cov}}}{1+c^-_{\text{cov}}}\|\mathbf{z}\|^2 > 0 \tag{9}$$

as otherwise positive definiteness of the covariance matrix would be lost. Moreover, iterations where the value on the left hand side of the inequality is close to zero may result in unstable behaviour as the covariance matrix would change rapidly. In order to prevent these problems, in iterations where long mutation vectors $\mathbf{z}$ lead to the left hand side of the inequality taking on a value of less than 0.5 we cap the value of c^-_{cov} at $1/(2\|\mathbf{z}\|^2 - 1)$ when computing the values of the coefficients in Eqs. (7) and (8).

Altogether, the $(1 + 1)$-CMA-ES with active covariance matrix adaptation differs from the algorithm given in Section 2 only in that the following step is added:

4. If $\mathbf{y}$ is inferior to its fifth-order ancestor, then update $\mathbf{A}$ and $\mathbf{A}_{\text{inv}}$ according to Eqs. (4) and (5) with the coefficients computed according to Eqs. (7) and (8) and c^-_{cov} clamped at $1/(2\|\mathbf{z}\|^2 - 1)$ if $1 < c^-_{\text{cov}}(2\|\mathbf{z}\|^2 - 1)$.[5]

The values of the constants used in the algorithm are summarised in Tab. 1. The settings for d, c, c_P, c^+_{cov}, and P_{target} are identical to the default values in [11]. The setting for c^-_{cov} has been obtained by running experiments on test functions of search space dimensionalities ranging from $n = 2$ to $n = 40$, in each instance choosing a suitable value, and fitting a regression curve through the resulting data points. It represents a compromise between larger values, which would allow faster adaptation on some functions, such as the discus function discussed below, and the desire not to lose performance on other functions where active covariance matrix adaptation is not beneficial and even detrimental with too large a coefficient, such as the sphere and cigar functions.

4. EXPERIMENTAL EVALUATION

In order to systematically evaluate the benefits of active covariance matrix adaptation for the $(1 + 1)$-CMA-ES we compare it with the corresponding strategy that does not

[5]Like the update of the search path in Section 2, the update of the Cholesky factors is stalled if the current success probability estimate P_{succ} exceeds P_{thresh}.

Table 2: Test functions.

sphere	$f_{\text{sphere}}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$
ellipsoid	$f_{\text{ellipsoid}}(\mathbf{x}) = \sum_{i=1}^{n} s^{\frac{i-1}{n-1}} x_i^2$
cigar	$f_{\text{cigar}}(\mathbf{x}) = x_1^2 + \sum_{i=2}^{n} s x_i^2$
discus	$f_{\text{discus}}(\mathbf{x}) = s x_1^2 + \sum_{i=2}^{n} x_i^2$
cigar-discus	$f_{\text{cigdis}}(\mathbf{x}) = s x_1^2 + \sum_{i=2}^{n-1} s^{\frac{1}{2}} x_i^2 + x_n^2$
two-axes	$f_{\text{two-axes}}(\mathbf{x}) = \sum_{i=1}^{\lfloor \vartheta n \rfloor} s x_i^2 + \sum_{i=\lfloor \vartheta n \rfloor + 1}^{n} x_i^2$
diff. powers	$f_{\text{diffpow}}(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert^{2+10\frac{i-1}{n-1}}$
Rosenbrock	$f_{\text{Rosen}}(\mathbf{x}) = \sum_{i=1}^{n-1} \left(100\left(x_i^2 - x_{i+1}\right)^2 + \left(x_i - 1\right)^2\right)$

use active covariance matrix adaptation. Section 4.1 describes results observed on a set of well-understood, convex-quadratic functions that are frequently used to evaluate the performance of real-valued evolutionary algorithms. Section 4.2 considers non-quadratic test functions.

4.1 Convex-Quadratic Functions

The first six of the test functions in Tab. 2 are convex-quadratic and have been employed previously for example in [6]. In all cases, the scaling factor s is set to 10^6, resulting in mostly ill-conditioned problems. Notice that while the functions are separable, this is not a limitation as the (1+1)-CMA-ES is invariant with regard to rigid transformations of the coordinate system. Applying random rotations would result in non-separable functions without affecting the results.

All runs are initialised with parental candidate solution $\mathbf{x}$ drawn from an n-dimensional standard normal distribution centred at the origin. The initial global step size σ is set to 0.1, the covariance matrix is set to the identity matrix, and the search path is initialised to the zero vector. All runs are terminated once a candidate solution with an objective function value of $f_{\text{stop}} = 10^{-10}$ or better has been generated.

The behaviour of CMA-ES on convex-quadratic functions is comparatively well understood. The optimal covariance matrix equals (a scalar multiple of) the inverse of the Hessian matrix, which is constant throughout the search space. Given enough time, CMA-ES learn a covariance matrix close to the optimal one and then proceed as fast as they would on the sphere function. With f_{stop} as small as considered here, it is largely the amount of time required to learn an approximation to the inverse Hessian that determines the number of time steps required to satisfy the termination condition for moderate and high dimensional problems.

To compare the performance of the strategy that uses active covariance matrix adaptation with the one that does not, we perform k_n runs with each of the algorithms for every n-dimensional test function. The number of runs k_n ranges from 10,000 for $n = 2$ down to 100 for $n = 40$. The number of time steps required to terminate has a greater coefficient of variation in lower-dimensional search spaces, necessitating the larger number of runs. We divide the median number of function evaluations required in runs of the strategy employing active covariance matrix adaptation by the median number of evaluations used by the original strategy that does not use active covariance matrix adaptation and refer to the result as the median runtime ratio. Median runtime ratio values of less than 1.0 indicate that active covariance matrix adaptation is beneficial, while values exceeding 1.0 indicate that it slows down the strategy.

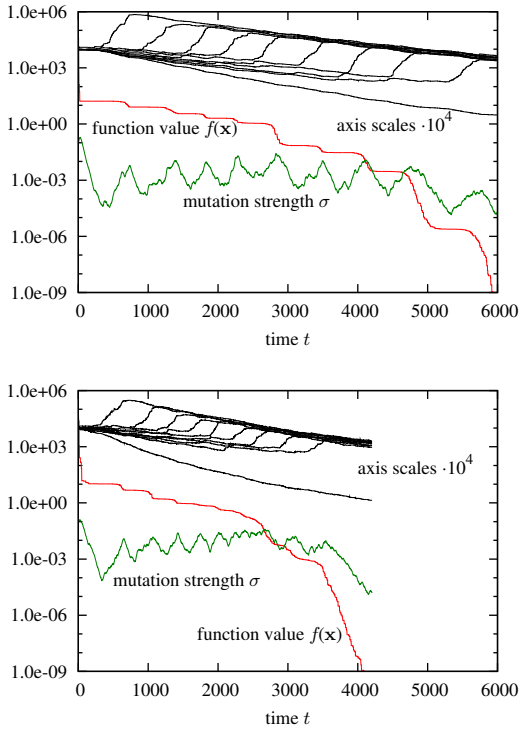

Figure 1: Typical runs of the $(1+1)$-CMA-ES on a ten-dimensional discus function without (top) and with (bottom) active covariance matrix adaptation.

Figure 1 illustrates the behaviour of the (1+1)-CMA-ES in typical runs on the discus function with $n = 10$. Shown are the objective function value of the search point, the global step size σ, and the square roots of the eigenvalues of the covariance matrix $\mathbf{C}$ (labelled as "axis scales"), multiplied with 10^4 for clarity. Notice that the latter are included for illustrative purposes and would not normally be computed by the $(1+1)$-CMA-ES as their computation requires $\Theta(n^3)$ time. The discus function is characterised by one of the eigenvalues of its Hessian matrix being significantly larger than all of the remaining ones. The optimal covariance matrix thus has one eigenvalue significantly smaller than the others. The axis scales in Fig. 1 suggest that a covariance matrix close to the optimal one is achieved toward the end of the runs as nine eigenvalues have a similar magnitude while the remaining one is smaller by a factor of about $10^6 = s$. Comparing the two subfigures, active covariance matrix adaptation allows reducing the magnitude of the single eigenvalue faster than the passive decay mechanism in the original strategy does. The speed-up resulting from active covariance matrix adaptation is almost one third in this case. However, due to the nature of its eigenvalue spectrum the discus function is a function for which active covariance matrix adaptation can be expected to be especially beneficial.

Figure 2 shows the number of time steps required to reach f_{stop} for all convex-quadratic test functions and search space dimensionalities ranging from $n = 2$ (top) to $n = 40$ (bottom). For each function, the left hand bars represent median values for the original $(1 + 1)$-CMA-ES, the right hand bars those for the strategy using active covariance matrix adaptation. The whiskers mark the 10th and 90th percentiles. The numbers given in parentheses in the figure are the respective median runtime ratio values.

For $n = 2$, active covariance matrix adaptation is beneficial for all of the functions. The median speed-up is 5% for the sphere function and 13% for all of the other functions (which are identical for $n = 2$). However, with increasing search space dimensionality the picture changes, and different behaviours can be observed for different groups of functions:

- For the sphere and cigar functions, active covariance matrix adaptation is either of little benefit for larger values of n, or it even results in a slight slow-down of the strategy. However, in the worst case the loss in performance is only about 3%.
- On $f_{\text{two-axes}}$ (here with $\vartheta = 0.5$) and on the ellipsoid function the benefits of active covariance matrix adaptation appear to diminish with increasing n. In no instance do the savings in the median number of steps exceed 13%, and for $f_{\text{two-axes}}$ the speed-up turns into a slight slow-down for $n = 40$.
- Active covariance matrix adaptation results in significant benefits across all search space dimensionalities on f_{cigdis} and, as expected, on f_{discus} where a speed-up by up to 46% can be observed. Experiments in higher-dimensional search spaces show that on the discus function the performance gap between the strategy that uses active covariance matrix adaptation and the one that does not continues to widen. For both f_{discus} and f_{cigdis}, active covariance matrix adaptation allows rapidly reducing the eigenvalue of the covariance matrix that corresponds to the direction where large variances are detrimental and lead to unsuccessful candidate solutions.

The behaviour of active covariance matrix adaptation on $f_{\text{two-axes}}$ is investigated further in Fig. 3, where the parameter ϑ that determines the percentage of directions that have a large eigenvalue associated with them is varied. This class of functions has been considered by Arnold [1] for strategies relying on isotropically distributed mutation vectors. For $\vartheta = 1/n$, $f_{\text{two-axes}}$ is the discus function. As ϑ increases, the dimensionality of the valley spanned by the eigenvectors associated with small eigenvalues decreases. For $\vartheta = (n-1)/n$, $f_{\text{two-axes}}$ is the cigar function, and at $\vartheta = 1$ the sphere function is reached. It can be seen that the benefits resulting from the use of active covariance matrix adaptation decrease gradually as the dimensionality of the valley spanned by the eigenvectors associated with small eigenvalues increases.

4.2 Non-Quadratic Functions

The convex-quadratic functions considered thus far are useful for their uniform characteristics across search space dimensionalities as well as for being well understood. However, clearly, they do not capture all aspects of general optimisation problems. We consider several non-quadratic test

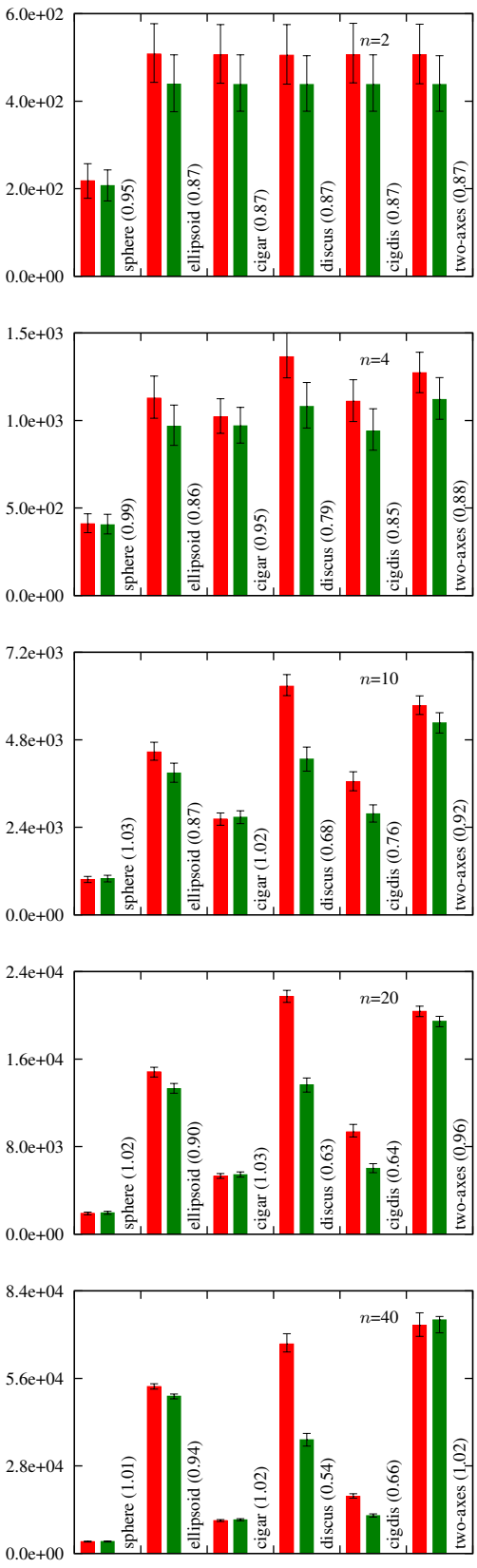

Figure 2: Number of time steps without (left) and with active covariance matrix adaptation (right).

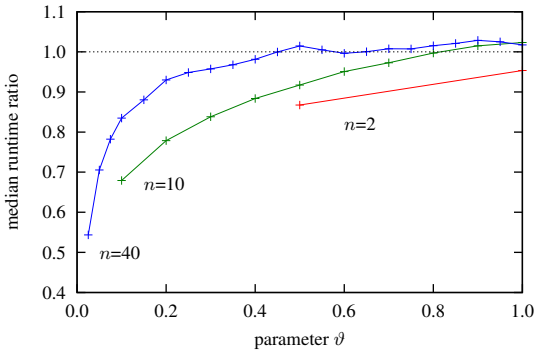

Figure 3: Median runtime ratios resulting from the use of active covariance matrix adaptation for $f_{\text{two-axes}}$ with varying ϑ.

functions from the set compiled by Moré, Garbow, and Hillstrom [9] that require continued adaptation of the covariance matrix throughout the run of the strategy. Those functions are of the form

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i^2(\mathbf{x})$$

with details gathered in Tab. 3. The first six functions are low-dimensional; the remaining three are of variable dimensionality. The Moré, Garbow, and Hillstrom test set is most commonly used to evaluate mathematical optimisation strategies. A complication arises when using them as test functions for evolutionary algorithms. The starting points in [9] are chosen such that a gradient based algorithm will follow the "right" valley to ultimately arrive at the globally optimal solution. Evolutionary algorithms are more exploratory in nature and occasionally end up in local minima different from the global ones.[6] Our choice of a subset of the test functions in [9] was motivated by the wish to exclude functions where the CMA-ES frequently reaches stationary points that are not globally optimal. Among the functions not excluded by that criterion, we have constrained ourselves to those with an optimal function value of 0.0 in order not to have to adapt our termination criterion, and we have preferred functions that have concise representations and do not depend on further parameters. No attempt has been made to filter the test set based on the performance of the optimisation strategies.

In addition to the non-quadratic test functions from [9], we also consider the different powers function and the generalised Rosenbrock function from Tab. 1. Both have been used extensively in performance evaluations of real-valued evolutionary algorithms. For the generalised Rosenbrock function, the existence of a local optimum different from the global one for $n \geq 4$ is well documented. As global optimisation is beyond the scope of this paper, when a run is found to converge to that local optimum it is aborted and repeated. This usually happens in fewer than 50% of runs. For all of the functions, the initialisation conditions and termination criterion are the same as in Section 4.1, except that for the test functions from [9] the initial search point is as

[6] See `http://www.uni-graz.at/imawww/kuntsevich/solvopt/results/moreset.html` for several local optima that have been identified.

Table 3: Test function details.

Powell badly scaled function: ($n = m = 2$, $\mathbf{x}_{\text{init}} = (0, 1)$)

$f_1(\mathbf{x}) = 10^4 x_1 x_2 - 1$

$f_2(\mathbf{x}) = \exp(-x_1) + \exp(-x_2) - 1.0001$

Brown badly scaled function: ($n = 2$, $m = 3$, $\mathbf{x}_{\text{init}} = (1, 1)$)

$f_1(\mathbf{x}) = x_1 - 10^6$

$f_2(\mathbf{x}) = x_2 - 2 \cdot 10^{-6}$

$f_3(\mathbf{x}) = x_1 x_2 - 2$

Beale function: ($n = 2$, $m = 3$, $\mathbf{x}_{\text{init}} = (1, 1)$)

$f_1(\mathbf{x}) = 1.5 - x_1(1 - x_2)$

$f_2(\mathbf{x}) = 2.25 - x_1(1 - x_2^2)$

$f_3(\mathbf{x}) = 2.625 - x_1(1 - x_2^3)$

helical valley function: ($n = m = 3$, $\mathbf{x}_{\text{init}} = (-1, 0, 0)$)

$$f_1(\mathbf{x}) = \begin{cases} 10(x_3 - 10\arctan(x_2/x_1)/(2\pi)) & \text{if } x_1 > 0 \\ 10(x_3 - 10\arctan(x_2/x_1)/(2\pi) - 5) & \text{otherwise} \end{cases}$$

$f_2(\mathbf{x}) = 10((x_1^2 + x_2^2)^{0.5} - 1)$

$f_3(\mathbf{x}) = x_3$

Powell singular function: ($n = m = 4$, $\mathbf{x}_{\text{init}} = (3, -1, 0, 1)$)

$f_1(\mathbf{x}) = x_1 + 10x_2$ $\qquad$ $f_3(\mathbf{x}) = (x_2 - 2x_3)^2$

$f_2(\mathbf{x}) = 5^{0.5}(x_3 - x_4)$ $\qquad$ $f_4(\mathbf{x}) = 10^{0.5}(x_1 - x_4)^2$

Wood function: ($n = 4$, $m = 6$, $\mathbf{x}_{\text{init}} = (-3, -1, -3, -1)$)

$f_1(\mathbf{x}) = 10(x_2 - x_1^2)$ $\qquad$ $f_4(\mathbf{x}) = 1 - x_3$

$f_2(\mathbf{x}) = 1 - x_1$ $\qquad$ $f_5(\mathbf{x}) = 10^{0.5}(x_2 + x_4 - 2)$

$f_3(\mathbf{x}) = 90^{0.5}(x_4 - x_3^2)$ $\qquad$ $f_6(\mathbf{x}) = 10^{-0.5}(x_2 - x_4)$

variably dimensioned function: ($m = n + 2$, $\mathbf{x}_{\text{init}} = (\xi_j)$ where $\xi_j = 1 - j/n$)

$f_i(\mathbf{x}) = x_i - 1$ $\qquad$ for $i = 1, \ldots, n$

$$f_{n+1}(\mathbf{x}) = \sum_{j=1}^{n} j(x_j - 1)$$

$$f_{n+2}(\mathbf{x}) = \left(\sum_{j=1}^{n} j(x_j - 1)\right)^2$$

Brown almost-linear func.: ($m = n$, $\mathbf{x}_{\text{init}} = (0.5, \ldots, 0.5)$)

$$f_i(\mathbf{x}) = x_i + \sum_{j=1}^{n} x_j - n - 1 \qquad \text{for } i = 1, \ldots, n-1$$

$$f_n(\mathbf{x}) = \left(\prod_{j=1}^{n} x_j\right) - 1$$

discrete boundary value function: ($m = n$, $\mathbf{x}_{\text{init}} = (\xi_j)$ where $\xi_j = t_j(t_j - 1)$, $t_j = jh$, and $h = 1/(n+1)$)

$f_i(\mathbf{x}) = 2x_i - x_{i-1} - x_{i+1} + h^2(x_i + t_i + 1)^3/2$

where $x_0 = x_{n+1} = 0$

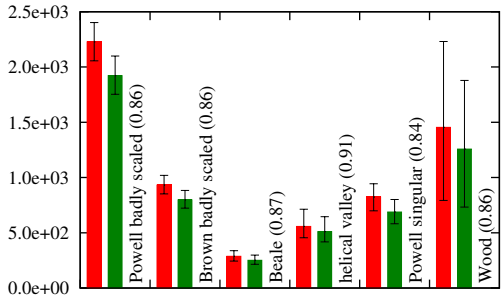

Figure 4: Number of time steps without (left) and with active covariance matrix adaptation (right).

indicated in Tab. 3 rather than being drawn from a normal distribution.

Figure 4 shows the number of function evaluations along with median runtime ratio values resulting from the use of active covariance matrix adaptation for the first six of the test functions from Tab. 3. The format of the graph is the same as in Fig. 2. The dimensionality of the functions ranges from $n = 2$ to $n = 4$. Median runtime ratio values are between 9% and 16% and thus in line with the values observed in Section 4.1 for convex-quadratic functions of similar dimensionalities. In all cases active covariance matrix adaptation is beneficial.

Figure 5 shows the corresponding results for the remaining three test functions from [9] as well as for f_{diffpow} and f_{Rosen} for search spaces dimensionalities ranging from $n = 2$ to $n = 40$. Again, the picture is not out of line with the results observed on convex-quadratic functions in Section 4.1. On the discrete boundary value function, as on the sphere and cigar functions, the speed-up resulting from the use of active covariance matrix is negligible. On the remaining functions, significant speed-ups that appear to increase with increasing search space dimensionality and some of which even surpass those on the discus function can be observed. In no case does active covariance matrix adaptation result in a loss of performance.

5. DISCUSSION AND CONCLUSIONS

In this paper we have introduced active covariance matrix adaptation for the (1+1)-CMA-ES. The algorithm retains all invariance properties of the original $(1+1)$-CMA-ES as well as its ability to avoid computationally expensive matrix decompositions. Whether active covariance matrix adaptation results in a performance advantage depends on the nature of the objective function. Significant benefits can be observed on convex-quadratic functions that are characterised by a small number of eigenvalues of their Hessian matrices that are significantly larger than the remaining ones. The eigenvectors associated with those eigenvalues span a set of directions in which objective function values vary rapidly. Active covariance matrix adaptation enables the evolution strategy to quickly reduce the variance of the mutation vectors in those directions and can result in a significant speed-up. On convex-quadratic functions without that characteristic, performance advantages are either much more limited, or a loss in performance not exceeding 3% can be observed. Experiments on non-quadratic test functions from the set of Moré,

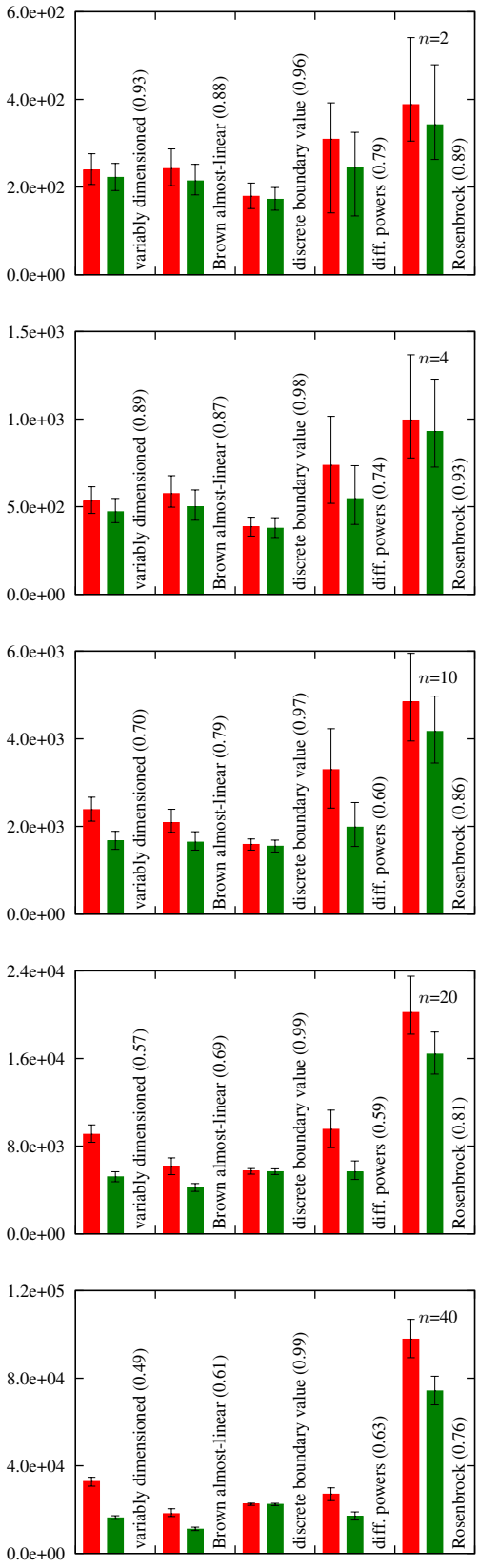

Figure 5: Number of time steps without (left) and with active covariance matrix adaptation (right).

Garbow, and Hillstrom have resulted in similar observations, with speed-up factors ranging from zero for some functions to savings up to about 50% for others.

Several opportunities may exist for achieving further improved performance of the algorithm. Normalising the especially unsuccessful vectors in Eq. (6) would eliminate the problem of long mutation vectors contributing over-proportionally to negative covariance matrix updates due to their length rather than their direction. A further minor tweak might be to make the value of c_{cov}^{-} dependent on the number of its immediate ancestors that it is inferior to.

Considering the objectives of speed, simplicity, and robustness, the $(1+1)$-CMA-ES with active covariance matrix adaptation is not dominated by any evolutionary algorithm. By extension of the observations in [7], it replaces the original $(1+1)$-CMA-ES as the fastest evolution strategy for unimodal optimisation problems. Its simplicity and robustness result from the ability to proceed without matrix decompositions and its invariance properties. There are several scenarios where the $(1+1)$-CMA-ES with active covariance matrix adaptation presents itself as a useful candidate optimisation algorithm:

- For unimodal and smooth functions, $(\mu/\mu, \lambda)$-CMA-ES have been observed by Auger et al. [4] to outperform an implementation of the BFGS algorithm for severely ill-conditioned problems, where the latter strategy suffers from numerical issues. The $(1+1)$-CMA-ES outperforms population-based evolution strategies in this scenario, and its ability to proceed without matrix decompositions makes it particularly useful if the search space dimensionality is high and objective function evaluations are relatively cheap.
- As evolutionary algorithms make no attempt to compute gradient vectors, they may be useful for non-smooth optimisation. A systematic comparison with algorithms for non-smooth optimisation, such as pattern search strategies, remains to be done.
- Severely noisy problems are better optimised using population based strategies that benefit from using larger step lengths, resulting in the ability to implicitly filter the objective function. However, work by Arnold and Beyer [2] has shown the potential benefits of overvaluation in elitist strategies, and the $(1+1)$-CMA-ES may be an appropriate algorithm for problems with low levels of noise.
- Very rugged functions with many local optima may resemble situations with large amounts of noise present and are thus usually better solved using population based algorithms. However, the relatively fast convergence of the $(1+1)$-CMA-ES may put it at an advantage in scenarios where there are relatively few local optima, due to the ability to perform a greater number of restarts.

Experimental research for each of these scenarios will be the subject of future work.

ACKNOWLEDGEMENTS

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

6. REFERENCES

[1] D. V. Arnold. On the use of evolution strategies for optimising certain positive definite quadratic forms. In *Genetic and Evolutionary Computation Conference — GECCO 2007*, pages 634–641. ACM Press, 2007.

[2] D. V. Arnold and H.-G. Beyer. Local performance of the $(1+1)$-ES in a noisy environment. *IEEE Transactions on Evolutionary Computation*, 6(1):30–41, 2002.

[3] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *IEEE Congress on Evolutionary Computation — CEC 2005*, pages 1769–1776. IEEE Press, 2005.

[4] A. Auger, N. Hansen, J. M. Perez Zerpa, R. Ros, and M. Schoenauer. Experimental comparisons of derivative free optimization algorithms. In J. Vahrenhold, editor, *Proc. of the 8th International Symposium on Experimental Algorithms*, pages 3–15. Springer Verlag, 2009.

[5] H.-G. Beyer and H.-P. Schwefel. Evolution strategies — A comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

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

[7] C. Igel, T. Suttorp, and N. Hansen. A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In *Genetic and Evolutionary Computation Conference — GECCO 2006*, pages 453–460. ACM Press, 2006.

[8] G. A. Jastrebski and D. V. Arnold. Improving evolution strategies through active covariance matrix adaptation. In *IEEE World Congress on Computational Intelligence — WCCI 2006*, pages 9719–9726. IEEE Press, 2006.

[9] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, pages 17–41, 1981.

[10] R. Ros and N. Hansen. A simple modification in CMA-ES achieving linear time and space complexity. In G. Rudolph et al., editors, *Parallel Problem Solving from Nature — PPSN X*, pages 296–305. Springer Verlag, 2008.

[11] T. Suttorp, N. Hansen, and C. Igel. Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning*, 75(2):167–197, 2009.