



A (1+1)-CMA-ES for Constrained Optimisation

Dirk V. Arnold, Nikolaus Hansen

► To cite this version:

Dirk V. Arnold, Nikolaus Hansen. A (1+1)-CMA-ES for Constrained Optimisation. GECCO, ACM, Jul 2012, Philadelphia, United States. pp.297-304, 10.1145/2330163.2330207 . hal-00696268

HAL Id: hal-00696268

<https://inria.hal.science/hal-00696268>

Submitted on 11 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A (1+1)-CMA-ES for Constrained Optimisation

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

This paper introduces a novel constraint handling approach for covariance matrix adaptation evolution strategies (CMA-ES). The key idea is to approximate the directions of the local normal vectors of the constraint boundaries by accumulating steps that violate the respective constraints, and to then reduce variances of the mutation distribution in those directions. The resulting strategy is able to approach the boundary of the feasible region without being impeded in its ability to search in directions tangential to the boundaries. The approach is implemented in the (1 + 1)-CMA-ES and evaluated numerically on several test problems. The results compare very favourably with data for other constraint handling approaches applied to unimodal test problems that can be found in the literature.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Stochastic optimisation, constraint handling, variable metric algorithm, evolution strategy

1. INTRODUCTION

Numerous constraint handling approaches have been proposed for use in evolutionary algorithms (EAs). A list of references¹ to constraint handling techniques used with EAs compiled by Coello Coello at the time of this writing contains more than one thousand entries. Among the most com-

monly employed approaches are algorithms that rely on immediate resampling of infeasible candidate solutions, penalty functions, or multiobjective techniques. A recent survey [15] provides a comprehensive overview.

Despite their strong performance in black box optimisation benchmarking exercises focusing on unconstrained real-valued problems [6], little research has been done on constraint handling techniques for covariance matrix adaptation evolution strategies (CMA-ES). The objective of this paper is to develop a simple and robust constraint handling approach for use in CMA-ES, and to incorporate it in the (1 + 1)-CMA-ES. We consider the problem of minimising $f(\mathbf{x})$ subject to inequality constraints $g_j(\mathbf{x}) \leq 0$ for $j = 1, \dots, m$. Function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is referred to as the objective function; functions $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are the constraint functions. In contrast to many existing constraint handling approaches, we do not make the assumption that it is possible to obtain meaningful objective function values for infeasible candidate solutions. Moreover, our strategy does not make use of constraint function values other than for the purpose of determining feasibility of a candidate solution. That is, even though the constraint functions yield real values, our algorithm only makes use of the binary information whether a constraint is violated or not. As a result, our approach enjoys invariance properties that algorithms that rely on quantifying the degree of constraint violation do not. See Hansen et al. [7] for a discussion of the importance of invariance properties in black box optimisation.

The key idea underlying our algorithm is to approximate the directions of the normal vectors of the constraint boundaries in the vicinity of the current candidate solution by accumulating steps that violate the respective constraints. Subsequently reducing variances of the distribution of offspring candidate solutions in those directions enables the strategy to closely approach the boundary of the feasible region without inhibiting its ability to search in orthogonal directions.

The remainder of this paper is organised as follows. Section 2 discusses constraint handling techniques used in connection with evolution strategies and then summarises the main ideas underlying the (1+1)-CMA-ES for unconstrained optimisation. Section 3 introduces the novel constraint handling approach that is the subject of this paper and describes its incorporation in the (1 + 1)-CMA-ES. Section 4 experimentally evaluates the algorithm on several test functions and compares its performance with that of other EAs for constrained optimisation. Section 5 concludes with a discussion of the results and directions for future research.

¹<http://www.cs.cinvestav.mx/~constraint/>

2. BACKGROUND

This section first reviews constraint handling techniques proposed in connection with evolution strategies. It then provides a brief review of the $(1+1)$ -CMA-ES, which forms the basis of the approach introduced in Section 3.

2.1 Related Work

Two of the simplest constraint handling techniques used in connection with evolution strategies are discussed by Oyman et al. [17]. One approach resamples infeasible candidate solutions until a feasible one has been generated. The other approach prefers feasible candidate solutions over infeasible ones, and it prefers those candidate solutions with smaller overall degrees of constraint violation in comparisons among infeasible candidate solutions. Neither approach requires computing objective function values for infeasible candidate solutions. In the case of the $(1+1)$ -ES, which adapts its step size based on observed success probabilities, the difference between the two approaches is highly significant. Fig. 1(a) illustrates a linear problem with a linear constraint. We refer to the angle between the normal vector of the constraint boundary and the gradient vector of the objective function as the constraint angle. If the parental candidate solution is in close proximity to the constraint plane, then small constraint angles result in a small probability of generating successful offspring. The approach that resamples infeasible candidate solutions does not consider those in the calculation of the success probability and thus does not reduce the step size in response to infeasible candidate solutions. In situations where there are multiple constraints active at the same time, this may result in generating very large numbers of infeasible candidate solutions before a feasible one is found. The approach that ranks feasible candidate solutions higher than infeasible ones does consider infeasible candidate solutions in its calculation of the success probability and systematically reduces the step size, resulting in convergence to a non-stationary point of the objective. This issue was first pointed out by Schwefel [19] and has since been studied analytically by Arnold and Brauer [1]. Fig. 1(b) illustrates that small constraint angles may arise in connection with nonlinear constrained optimisation problems.

Runarsson and Yao [18] propose stochastic ranking as an approach for weighting differences in objective function values and constraint violations against each other. The approach requires the ability to obtain objective function values for infeasible candidate solutions.

Kramer et al. [14] propose the use of biased mutations for solving constrained optimisation problems. They do not evaluate their approach in an evolution strategy that is invariant with regard to rotations of the coordinate system.

Kramer and Schwefel [13] propose and experimentally evaluate several constraint handling techniques for evolution strategies, ranging from adaptively lowering a lower bound on the global step size of the strategy to an algorithm evolving populations of feasible and infeasible candidate solutions in tandem and one that involves a nested strategy that adapts rotation angles of the mutation ellipsoid. All of those have in common that no objective function values are computed for infeasible candidate solutions.

Kramer et al. [12] are the first to propose a constraint handling approach specifically for CMA-ES. Similar to our approach, they learn a model of the constraint function and use that model to adapt the covariance matrix that governs

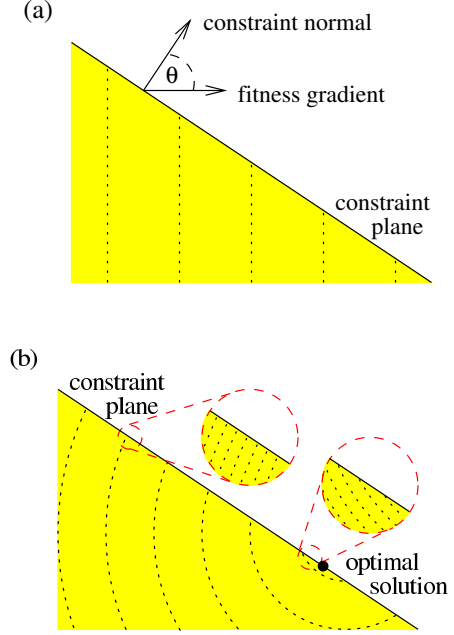


Figure 1: (a) Linear problem with a single linear constraint. The two-dimensional subspace spanned by the normal vector of the constraint plane and the gradient vector of the objective function is shown. The feasible region is shaded. (b) Two-dimensional quadratic problem with a linear constraint where the optimal solution lies on the boundary of the feasible region. Constraint angles decrease as the optimal solution is approached.

the distribution of offspring candidate solutions. However, both the nature of the model for the constraint functions and the covariance matrix update differ. Our model is simpler and obtained on the fly, without the need for additional constraint function evaluations. The covariance matrix update in [12] aims to rotate the covariance matrix, whereas ours acts to reduce variances in directions where constraint violations have been observed.

Collange et al. [3] introduce a constraint handling approach for CMA-ES that strives to ensure that within some given number of iterations the population contains at least one feasible candidate solution. Their approach relies on constraint function values and a user defined constraint value threshold. No results from an evaluation of the approach on test functions are provided.

2.2 (1+1)-CMA-ES

The $(1+1)$ -CMA-ES introduced by Igel et al. [10] is a variant of the $(1+1)$ -ES that adapts the entire covariance matrix of its offspring distribution in addition to its global step size. As the covariance matrix \mathbf{C} is positive definite, Cholesky decomposition yields $n \times n$ matrix \mathbf{A} such that $\mathbf{C} = \mathbf{A}\mathbf{A}^T$, and offspring candidate solutions \mathbf{y} are generated as

$$\mathbf{y} = \mathbf{x} + \sigma \mathbf{A} \mathbf{z} \quad (1)$$

where \mathbf{x} is the parental candidate solution, $\mathbf{z} \in \mathbb{R}^n$ has standard normally distributed components, and σ is the global step size of the strategy. The strategy maintains an exponentially fading record $\mathbf{s} \in \mathbb{R}^n$ of successful steps that is

referred to as the search path. Upon generating a successful offspring candidate solution it performs covariance matrix update

$$\mathbf{C} \leftarrow (1 - c_{\text{cov}}^+) \mathbf{C} + c_{\text{cov}}^+ \mathbf{s} \mathbf{s}^T \quad (2)$$

where $c_{\text{cov}}^+ \in (0, 1)$ is a constant. Rather than working with the covariance matrix and performing a Cholesky decomposition in every iteration of the algorithm, Igel et al. [10] show that equivalently, matrix \mathbf{A} can be updated according to

$$\begin{aligned} \mathbf{A} &\leftarrow \sqrt{1 - c_{\text{cov}}^+} \mathbf{A} \\ &+ \frac{\sqrt{1 - c_{\text{cov}}^+}}{\|\mathbf{w}\|^2} \left(\sqrt{1 + \frac{c_{\text{cov}}^+ \|\mathbf{w}\|^2}{1 - c_{\text{cov}}^+}} - 1 \right) \mathbf{s} \mathbf{w}^T \end{aligned} \quad (3)$$

where $\mathbf{w} = \mathbf{A}^{-1} \mathbf{s}$. Sutton et al. [20] provide an analogous update of the inverse of \mathbf{A} that removes the need for matrix inversion and makes it possible to implement the update with $\Theta(n^2)$ cost per iteration.

A variant of the $(1 + 1)$ -CMA-ES proposed by Arnold and Hansen [2] incorporates the active covariance matrix update due to Jastrebski and Arnold [11] into the strategy. The key observation is that in situations where there are isolated large eigenvalues of the Hessian matrix of the objective function, many iterations are required before the update in Eq. (2) yields appropriate step sizes in the directions of the corresponding eigenvectors. Rather than letting variances in those directions decay through repeated multiplication with $(1 - c_{\text{cov}}^+) < 1$, in situations where the offspring candidate solution is especially unsuccessful (in the sense that it is inferior to its k th order ancestor for some $k > 1$), the strategy performs covariance matrix update

$$\mathbf{C} \leftarrow (1 + c_{\text{cov}}^-) \mathbf{C} - c_{\text{cov}}^- (\mathbf{A} \mathbf{z})(\mathbf{A} \mathbf{z})^T \quad (4)$$

where $c_{\text{cov}}^- \in (0, 1)$ is a constant. Arnold and Hansen [2] show that the update can equivalently be accomplished by updating matrix \mathbf{A} according to

$$\begin{aligned} \mathbf{A} &\leftarrow \sqrt{1 + c_{\text{cov}}^-} \mathbf{A} \\ &+ \frac{\sqrt{1 + c_{\text{cov}}^-}}{\|\mathbf{z}\|^2} \left(\sqrt{1 - \frac{c_{\text{cov}}^- \|\mathbf{z}\|^2}{1 + c_{\text{cov}}^-}} - 1 \right) \mathbf{A} \mathbf{z} \mathbf{z}^T \end{aligned} \quad (5)$$

and they also provide an update of the inverse of \mathbf{A} that allows performing updates with $\Theta(n^2)$ cost per iteration.

3. ALGORITHM

The key idea underlying our constraint handling approach is to reduce variances of the distribution of offspring candidate solutions in the directions of the normal vectors of constraint boundaries in the vicinity of the current parental candidate solution. For that purpose, for each of the m constraints we maintain an exponentially fading record \mathbf{v}_j , $j = 1, \dots, m$, of steps that have violated the constraint. Those vectors are initialised to be zero and, in those iterations where the j th constraint is violated, updated according to

$$\mathbf{v}_j \leftarrow (1 - c_c) \mathbf{v}_j + c_c \mathbf{A} \mathbf{z} \quad (6)$$

We refer to the \mathbf{v}_j as constraint vectors. Accumulation acts as a low-pass filter and ensures that components of $\mathbf{A} \mathbf{z}$ that are tangential to the local constraint boundary average out

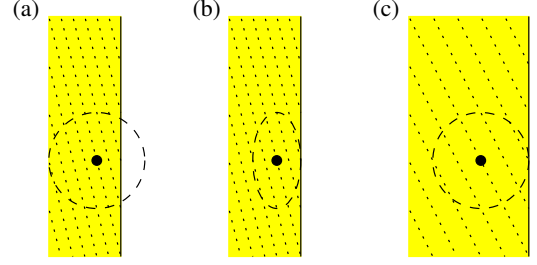


Figure 2: Effect of reducing the variance of the offspring distribution in the direction of the normal vector of the constraint boundary. The parental candidate solution is marked by a dot. (a) The original distribution is indicated by the dashed circle. Dotted lines are contour lines of the objective function. (b) The variance of the distribution in the horizontal direction is reduced. (c) Situation from (b) shown with the variables transformed by \mathbf{A}^{-1} .

in the mean. Parameter $c_c \in (0, 1)$ determines how quickly the information present in the constraint vectors fades.

In those iterations where the offspring candidate solution is infeasible, the Cholesky factor of the covariance matrix is updated according to

$$\mathbf{A} \leftarrow \mathbf{A} - \frac{\beta}{\sum_{j=1}^m \mathbb{1}_{g_j(\mathbf{y}) > 0}} \sum_{j=1}^m \mathbb{1}_{g_j(\mathbf{y}) > 0} \frac{\mathbf{v}_j \mathbf{w}_j^T}{\mathbf{w}_j^T \mathbf{w}_j} \quad (7)$$

where $\mathbf{w}_j = \mathbf{A}^{-1} \mathbf{v}_j$ and $\mathbb{1}_{g_j(\mathbf{y}) > 0}$ equals one if $g_j(\mathbf{y}) > 0$ and zero otherwise. The update is akin to that in Eq. (5) with the constraint vectors \mathbf{v}_j replacing $\mathbf{A} \mathbf{z}$. Parameter β in Eq. (7) controls the size of the updates similar to the way that c_{cov}^- does in Eq. (5). However, while the update in Eq. (5) is of rank one, that in Eq. (7) is a multiple-rank update if multiple constraints are violated. In contrast to the update in Eq. (5), there is no multiplication of matrix \mathbf{A} with a scalar greater than one in order to increase variances in directions other than those implicit in the constraint vectors. Also in contrast to the active covariance matrix update described in Section 2.2, we do not have an update of the inverse of matrix \mathbf{A} that would allow performing a step with cost quadratic in n . Instead, the computational cost of the update is in $\mathcal{O}(n^3 + mn^2)$.

Fig. 2 illustrates the effect of reducing the variance of the distribution of offspring candidate solutions in the direction of the normal vector of the constraint plane for a linear problem. Fig. 2(a) shows a parental candidate solution in the vicinity of the constraint boundary. The dashed circle indicates a contour line of the offspring density. Fig. 2(b) depicts the same situation with the variance of the offspring distribution reduced in the direction of the normal vector of the constraint plane as a result of the transformation implicit in matrix \mathbf{A} . Fig. 2(c) considers the same situation as in (b), but uses a coordinate system transformed by \mathbf{A}^{-1} . See Hansen [5] for a discussion of coordinate transformations and adaptive encodings. Notice that the change in offspring distribution results in an increase in the constraint angle in the transformed space in Fig. 2(c) compared to Fig. 2(a). If capable of learning an appropriate coordinate transformation, the algorithm will be able to avoid the pitfalls associated with small constraint angles discussed in Section 2.1.

1. Generate offspring candidate solution \mathbf{y} according to Eq. (1).
2. For $j = 1, \dots, m$, determine whether $g_j(\mathbf{y}) > 0$ and update \mathbf{v}_j according to Eq. (6) if it is.
3. If \mathbf{y} is infeasible, then compute $\mathbf{w}_j = \mathbf{A}^{-1}\mathbf{v}_j$ for all $j = 1, \dots, m$ and update the transformation matrix \mathbf{A} according to Eq. (7). The iteration is complete.
4. Otherwise, evaluate $f(\mathbf{y})$, update the success probability estimate according to

$$P_{\text{succ}} \leftarrow (1 - c_P)P_{\text{succ}} + c_P \mathbb{1}_{f(\mathbf{y}) \leq f(\mathbf{x})}$$

and update the global step size according to

$$\sigma \leftarrow \sigma \exp \left(\frac{1}{d} \frac{P_{\text{succ}} - P_{\text{target}}}{1 - P_{\text{target}}} \right).$$

5. If $f(\mathbf{y}) \leq f(\mathbf{x})$, then replace \mathbf{x} with \mathbf{y} , update the search path according to

$$\mathbf{s} \leftarrow (1 - c)\mathbf{s} + \sqrt{c(2 - c)}\mathbf{A}\mathbf{z}$$

and update matrix \mathbf{A} according to Eq. (3). The iteration is complete.

6. Otherwise, if $f(\mathbf{y})$ is inferior to its fifth order ancestor, update matrix \mathbf{A} according to Eq. (5).

Figure 3: Single iteration of the (1 + 1)-CMA-ES with constraint handling through active covariance matrix adaptation.

Incorporating the idea of constraint handling through active covariance matrix adaptation in the (1 + 1)-CMA-ES yields an algorithm the state of which is described by parental candidate solution \mathbf{x} along with its objective function value $f(\mathbf{x})$ and those of its five most recent ancestors, global step size σ , success probability estimate P_{succ} , transformation matrix \mathbf{A} , search path \mathbf{s} , and constraint vectors \mathbf{v}_j for $j = 1, \dots, m$. Each iteration of the algorithm updates those quantities as described in Fig. 3. For $\beta = 0$, the algorithm is identical to the (1 + 1)-CMA-ES as described in [2], with constraints handled using the simple resampling scheme described by Oyman et al. [17]. All parameter settings are summarised in Table 1. Values for parameters c_c and β have been obtained by optimising the performance of the strategy applied to linearly constrained sphere functions and then reducing β by a factor of about six in order to achieve robust performance on other test problems. All other parameter settings are identical to those given in [2].

4. EXPERIMENTAL EVALUATION

This section experimentally evaluates the performance of the algorithm proposed in Section 3 using several constrained test problems. Section 4.1 considers constrained sphere functions that serve to examine basic scaling properties. Section 4.2 considers several test problems frequently used to evaluate the performance of algorithms for constrained optimisation, and it compares results observed for the (1 + 1)-CMA-ES with constraint handling through active covariance matrix adaptation with published work.

Table 1: Parameter settings.

$$\begin{aligned} d &= 1 + \frac{n}{2} & c &= \frac{2}{n+2} & c_P &= \frac{1}{12} & P_{\text{target}} &= \frac{2}{11} \\ c_{\text{cov}}^+ &= \frac{2}{n^2 + 6} & c_{\text{cov}}^- &= \min \left(\frac{0.4}{n^{1.6} + 1}, \frac{1}{2\|\mathbf{z}\|^2 - 1} \right) \\ c_c &= \frac{1}{n+2} & \beta &= \frac{0.1}{n+2} \end{aligned}$$

4.1 Constrained Sphere Functions

Consider the problem of minimising function

$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

subject to constraints $x_i \geq 1$ for $i = 1, \dots, m \leq n$. The optimal solution has components $x_i = \mathbb{1}_{i \leq m}$ and objective function value m . Fig. 4 shows the number of objective and constraint function evaluations required to locate a solution with an objective function value that differs from the optimal one by no more than 10^{-8} plotted against the dimension of the search space. In the counting of constraint function evaluations it is assumed that one function call yields information for all constraints. The upper graph shows results for $m = 1$, the lower one for $m = n/2$.

Two variants of the (1 + 1)-CMA-ES are considered: one that employs constraint handling through active covariance matrix adaptation as described in Section 3, and one that relies on the simple resampling approach described by Oyman et al. [17] (which is recovered from the former for $\beta = 0$). In every run the initial candidate solution is sampled uniformly from the feasible subregion of $[-100, 100]^n$. The global step size σ and Cholesky factor \mathbf{A} are initialised to 0.1 and the $n \times n$ identity matrix, respectively. Runs are terminated when either the optimal solution has been found to within the desired accuracy, or when 10^6 iterations have failed to locate it. We have conducted 99 runs of either strategy in each case and ordered the runs by the number of objective function evaluations and constraint function calls. The bars in the figure indicate the 10th, 50th, and 90th smallest values observed. The thin dotted lines have slopes corresponding to linear and quadratic growth, respectively.

In the case of a single constraint, the strategy variant that employs active covariance matrix adaptation for constraint handling locates the optimal solution to within the desired accuracy in every run and for every value of n considered. Both the number of objective function evaluations and the number of constraint function calls appear to grow linearly with the dimension n . The number of objective function evaluations is less than twice that required on unconstrained sphere functions of the same dimension. The strategy variant that simply resamples infeasible candidate solutions locates the optimal solution to within the desired accuracy in every run up to $n = 40$. However, it requires more function evaluations, with the gap between the two variants widening as n increases. For $n = 100$, the resampling approach regularly fails to reach the termination condition within 10^6 iterations.

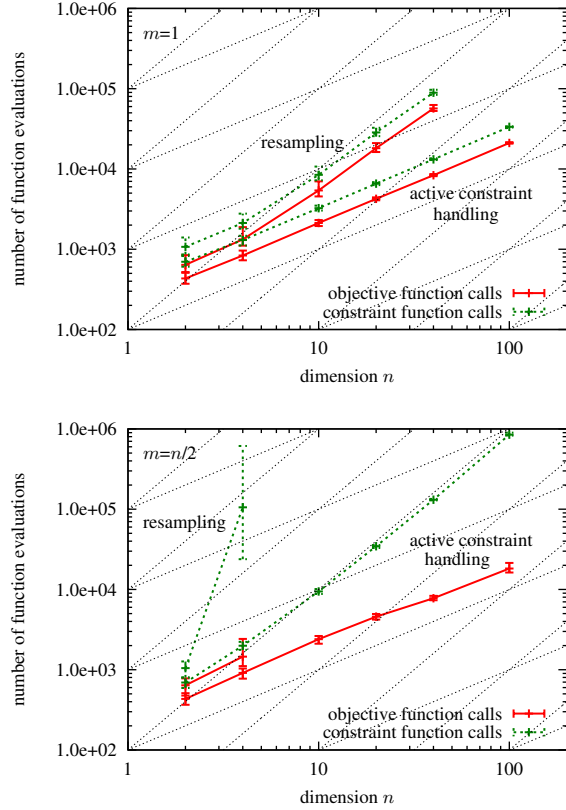


Figure 4: Comparison of the number of objective and constraint function evaluations required to optimise constrained sphere functions with $m = 1$ (top) and $m = n/2$ (bottom). The curves show results for the $(1+1)$ -CMA-ES with and without constraint handling through active covariance matrix adaptation.

In the case of $m = n/2$ constraints, the strategy variant that employs active covariance matrix adaptation for constraint handling again locates the optimal solution to within the desired accuracy in every run and for every value of n considered. The growth of the number of objective function evaluations again appears linear in n , with overall numbers very similar to those observed for $m = 1$. The growth of the number of constraint function calls appears to be quadratic in the dimension of the problem. Results for the strategy variant that resamples infeasible candidate solutions are incomplete as for $n \geq 10$ the strategy regularly fails to locate the optimal solution to within the desired accuracy.

4.2 Other Test Functions

In order to perform a comparison of our algorithm with published results, we have evaluated its performance on several test functions that are commonly used to evaluate EAs for constrained optimisation. Specifically, we have considered the test functions collected by Michalewicz and Schoenauer [16] as well as those employed by Kramer and Schwefel [13]. However, a comparison is difficult for several reasons:

- Many test problems have multiple local minima. The $(1+1)$ -ES is a poor global optimiser and not use-

fully applied to multimodal problems. We have restricted ourselves to test functions **g06**, **g07**, **g09**, and **g10** from the test suite in [16] as all others are multimodal. When comparing our results with those found in the literature, it is important to keep in mind that other EAs may sacrifice speed of convergence for improved global search performance, and that our choice of test functions is biased in favour of fast local optimisers. We expect that this caveat will be addressed as we incorporate the approach of handling constraints through active covariance matrix adaptation in the $(\mu/\mu, \lambda)$ -CMA-ES.

- Many algorithms for which results are reported in the literature compute objective function values for infeasible candidate solutions. While this is possible for commonly used artificial test problems, relying on the ability to obtain meaningful objective function values for infeasible candidate solutions limits the applicability of the algorithms for some practical problems. Presumably, being able to compute objective function values for infeasible candidate solutions can simplify the task of locating near optimal solutions. At the same time, those algorithms that evaluate infeasible candidate solutions incur computational costs that those that evaluate only feasible candidate solutions do not.
- Our algorithm requires a feasible candidate solution as starting point. Some test problems include recommended starting points, and we have used those. For those test problems where no starting point is given, we uniformly sample candidate solutions from the volume defined by the bound constraints until the first feasible candidate solution is encountered. Notice that no objective function evaluations are required for this process. The number of constraint function calls required to generate starting points is highly dependent on the choice of bound constraints and is not included in the figures given below. Algorithms that evaluate infeasible candidate solutions may have a bias in the first feasible candidate solutions that they encounter as the objective function landscape in the infeasible region may act as a funnel.
- Performance measures employed in different papers vary. Many authors only report objective function values after a number of objective function evaluations that far exceeds the computational budget required by our algorithm to locate near optimal solutions.

We have performed 99 runs of the $(1+1)$ -CMA-ES with constraint handling through active covariance matrix adaptation on each of the eight test problems described in Appendix A. For test problems **g06**, **g07**, **g09**, **g10**, and **HB**, runs were terminated when a solution was obtained that, when rounded to the accuracy with which the optimal values are given in Appendix A, was identical to the optimal value. For the remaining problems, runs were terminated when optimal objective function values were attained to within a factor of 10^{-8} . Table 2 lists the problems' dimensions n , the number m_{act} of constraints active at the location of the optimal solution, and the 10th, 50th, and 90th smallest numbers of objective and constraint function evaluations in each case. All runs have located the respective optimal solutions

Table 2: Function evaluations required to solve test problems.

problem	n	m_{act}	objective function			constraint function		
			10th	50th	90th	10th	50th	90th
g06	2	2	272	308	364	827	1060	1223
g07	10	6	1939	2211	2703	10435	11283	12704
g09	7	2	1430	1674	2074	3626	4106	5075
g10	8	6	2794	3976	5369	15621	18781	23088
TR2	2	1	376	443	510	616	708	839
2.40	5	5	1326	1990	3326	4551	6994	11114
2.41	5	5	1483	2271	3581	5235	8108	12056
HB	5	4	623	768	1150	2338	2912	3970

to within the desired accuracy. We have conducted analogous experiments with the strategy that employs the simple resampling scheme and observed failure to converge to the optimal solutions in many cases for all test problems but **g06** and **TR2**. In those cases where optimal solutions were attained, the number of constraint function calls was very significantly higher than for the strategy that employs active covariance matrix adaptation for constraint handling.

While a comparison with published work is difficult for the reasons outlined above and the figures need to be interpreted with care, some results from the literature are as follows:

- Runarsson and Yao [18] use stochastic ranking in a relatively basic evolution strategy that evaluates infeasible candidate solutions. They report the median number of function evaluations required to attain the best values encountered in each run. Those values are in some cases less accurate than those used as termination conditions here. For test problems **g06**, **g07**, **g09**, and **g10** they report figures of 108,000, 143,000, 124,000, and 128,000 objective and as many constraint function calls, respectively. These values exceed the median values reported in Table 2 by factors of 350.6, 64.7, 74.1, and 32.2 for objective function calls and by factors of 101.9, 12.7, 30.2, and 6.8 for constraint function calls.
- In a paper that was ranked first in the 2006 IEEE Congress on Evolutionary Computation Special Session on Constrained Real-Parameter Optimization, Takahama and Sakai [21] report the number of function calls required by ϵ constrained differential evolution with gradient-based mutation and feasible elites to obtain solutions within 10^{-4} of optimal solutions for **g06**, **g07**, **g09**, and **g10** to be 7,381, 74,303, 23,121, and 105,234 in the mean, respectively. These values exceed the median values reported in Table 2 by factors of 24.0, 33.6, 13.8, and 26.5 for objective function calls and by factors of 7.0, 6.6, 5.6, and 5.6 for constraint function calls.
- Takahama and Sakai [22] report to have reached optimal solutions for problems **g06**, **g07**, **g09**, and **g10** using ϵ constrained adaptive differential evolution after 15,846, 38,402, 25,400, and 24,815 objective function calls on average, respectively. These values exceed the median values reported in Table 2 by factors of 51.4, 17.4, 15.2, and 6.2. The corresponding numbers of constraint function calls are 36,112, 99,896, 50,766,

and 99,866 and exceed the median values reported in Table 2 by factors of 34.1, 8.9, 12.4, and 5.3.

- Kramer and Schwefel [13] apply evolution strategies with several constraint handling mechanisms to problems **TR2**, **2.40**, **2.41**, and **HB**. The best figures they report for any of the evolution strategies are 796,201, 79,566, 51,660, and 54,344 objective function calls and 1,100,872, 770,334, 379,268, and 211,499 constraint function calls, respectively. These values exceed the median values reported in Table 2 by factors of 1,797.3, 40.0, 22.7, and 70.8 for objective function calls and by factors of 1,554.9, 110.1, 46.8, and 72.6 for constraint function calls.
- Kramer et al. [12] employ a CMA-ES that learns constraint function models and rotates mutation distributions accordingly. They report figures of 3,249 and 11,216 objective function calls and 3,650 and 30,068 constraint function calls for problems **TR2** and **2.40**, respectively. These figures exceed the median numbers of objective function calls reported in Table 2 by factors of 7.3 and 5.6. The corresponding numbers of constraint function calls differ by factors of 5.2 and 4.3.

Keeping the caveats mentioned above in mind, these results suggest the $(1+1)$ -CMA-ES with constraint handling through active covariance matrix adaptation as a relatively capable EA for solving unimodal constrained optimisation problems.

5. DISCUSSION AND FUTURE WORK

We have proposed a novel approach for handling constraints in covariance matrix adaptation evolution strategies. Our algorithm does not assume that meaningful objective function values can be obtained for infeasible candidate solutions. It does not make use of constraint function values other than for determining feasibility of candidate solutions and thus has desirable invariance properties. The key idea of the approach is to obtain approximations to the directions of normal vectors of constraint boundaries in the vicinity of the location of current candidate solutions by low-pass filtering steps that violate the respective constraints, and to then reduce variances of the offspring distribution in those directions. The scaling implicit in the transformation that is learned in the process has the effect of increasing local constraint angles. We have presented experimental evidence that if incorporated in the $(1+1)$ -CMA-ES, the proposed constraint handling approach outperforms a simple resampling scheme. Further experiments on several unimodal test

problems suggest that the approach may be significantly more efficient than other approaches for constrained evolutionary optimisation.

In future work we will conduct comparisons of the performance of the evolution strategy introduced here with further algorithms for constrained optimisation, such as mesh adaptive direct search algorithms and generating set search methods. Furthermore, we will incorporate the idea of constraint handling through active covariance matrix adaptation in the $(\mu/\mu, \lambda)$ -CMA-ES and evaluate its potential for solving constrained multimodal optimisation problems.

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 and D. Brauer. On the behaviour of the $(1+1)$ -ES for a simple constrained problem. In G. Rudolph et al., editors, *Parallel Problem Solving from Nature — PPSN X*, pages 1–10. Springer Verlag, 2008.
- [2] D. V. Arnold and N. Hansen. Active covariance matrix adaptation for the $(1+1)$ -CMA-ES. In *Genetic and Evolutionary Computation Conference — GECCO 2010*, pages 385–392. ACM Press, 2010.
- [3] G. Collange, N. Delattre, N. Hansen, I. Quinquis, and M. Schoenauer. Multidisciplinary optimisation in the design of future space launchers. In P. Breitkopf and R. F. Coelho, editors, *Multidisciplinary Design Optimization in Computational Mechanics*, pages 487–496. Wiley, 2010.
- [4] C. Floudas and P. Pardalos. *A Collection of Test Problems for Constrained Global Optimization Algorithms*. Springer Verlag, 1987.
- [5] N. Hansen. Adaptive encoding: How to render search coordinate system invariant. In G. Rudolph et al., editors, *Parallel Problem Solving from Nature — PPSN X*, pages 205–214. Springer Verlag, 2008.
- [6] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Genetic and Evolutionary Computation Conference Companion — GECCO 2010*, pages 1689–1696. ACM Press, 2010.
- [7] N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger. Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11(8):5755–5769, 2011.
- [8] D. M. Himmelblau. *Applied Nonlinear Programming*. McGraw-Hill, 1972.
- [9] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Springer Verlag, 1981.
- [10] 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.
- [11] 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.
- [12] O. Kramer, A. Barthelmes, and G. Rudolph. Surrogate constraint functions for CMA evolution strategies. In B. Mertsching et al., editors, *KI 2009: Advances in Artificial Intelligence*, pages 169–176. Springer Verlag, 2009.
- [13] O. Kramer and H.-P. Schwefel. On three new approaches to handle constraints within evolution strategies. *Natural Computing*, 5(4):363–385, 2006.
- [14] O. Kramer, C.-K. Ting, and H. Kleine Büning. A new mutation operator for evolution strategies for constrained problems. In *IEEE Congress on Evolutionary Computation — CEC 2005*, pages 2600–2606. IEEE Press, 2005.
- [15] E. Mezura-Montes and C. A. Coello Coello. Constraint-handling in nature-inspired numerical optimization: Past, present, and future. *Swarm and Evolutionary Computation*, 1(4):173–194, 2011.
- [16] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32, 1996.
- [17] A. I. Oyman, K. Deb, and H.-G. Beyer. An alternative constraint handling method for evolution strategies. In *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pages 612–619. IEEE Press, 1999.
- [18] T. P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):274–283, 2000.
- [19] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, 1995.
- [20] T. Suttorp, N. Hansen, and C. Igel. Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning*, 75(2):167–197, 2009.
- [21] T. Takahama and S. Sakai. Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites. In *IEEE World Congress on Computational Intelligence — WCCI 2006*, pages 308–315. IEEE Press, 2006.
- [22] T. Takahama and S. Sakai. Efficient constrained optimization by the ϵ constrained adaptive differential evolution. In *IEEE World Congress on Computational Intelligence — WCCI 2010*, pages 2052–2059. IEEE Press, 2010.

APPENDIX

A. TEST PROBLEMS

Problem g06: (Floudas and Pardalos [4]) Minimise

$$f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

subject to

$$g_1(\mathbf{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$

$$g_2(\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$$

and bound constraints $13 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$. Both constraints other than the bound constraints

are active at the optimum, which has an objective function value of -6961.81381 .

Problem g07: (Hock and Schittkowski [9]) Minimise

$$f(\mathbf{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 \\ + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 \\ + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

subject to

$$g_1(\mathbf{x}) = 4x_1 + 5x_2 - 3x_7 + 9x_8 - 105 \leq 0 \\ g_2(\mathbf{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(\mathbf{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\ g_4(\mathbf{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \\ g_5(\mathbf{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\ g_6(\mathbf{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\ g_7(\mathbf{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\ g_8(\mathbf{x}) = (x_1 - 8)^2 + 4(x_2 - 4)^2 + 6x_5^2 - 2x_6 - 60 \leq 0$$

and bound constraints $-10 \leq x_i \leq 10$ for $i = 1, \dots, 10$. Six of the eight constraints other than the bound constraints are active at the optimum, which has an objective function value of 24.3062091 .

Problem g09: (Hock and Schittkowski [9]) Minimise

$$f(\mathbf{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\ + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to

$$g_1(\mathbf{x}) = -127 + 2x_1^2 + 3x_4^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\ g_2(\mathbf{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\ g_3(\mathbf{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\ g_4(\mathbf{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

and bound constraints $-10 \leq x_i \leq 10$ for $i = 1, \dots, 7$. Two of the constraints other than the bound constraints are active at the optimum, which has an objective function value of 680.630057 .

Problem g10: (Hock and Schittkowski [9]) Minimise

$$f(\mathbf{x}) = x_1 + x_2 + x_3$$

subject to

$$g_1(\mathbf{x}) = 0.0025(x_4 + x_6) - 1 \leq 0 \\ g_2(\mathbf{x}) = 0.0025(x_5 + x_7 - x_4) - 1 \leq 0 \\ g_3(\mathbf{x}) = 0.01(x_8 - x_5) - 1 \leq 0 \\ g_4(\mathbf{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0 \\ g_5(\mathbf{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\ g_6(\mathbf{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$$

and bound constraints $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$ for $i = 2, 3$, and $10 \leq x_i \leq 1000$ for $i = 4, \dots, 8$. All six constraints other than the bound constraints are active at the optimum, which has an objective function value of 7049.2480 .

Problem TR2: (Kramer and Schwefel [13]) Minimise

$$f(\mathbf{x}) = x_1^2 + x_2^2$$

subject to

$$g_1(\mathbf{x}) = 2 - x_1 - x_2 \leq 0.$$

The constraint is active at the optimum, which has an objective function value of 2 . Point $(50, 50)^T$ serves as the starting point.

Problem 2.40: (Schwefel [19]) Minimise

$$f(\mathbf{x}) = -\sum_{i=1}^5 x_i$$

subject to

$$g_1(\mathbf{x}) = \sum_{i=1}^5 (9 + i)x_i - 50000 \leq 0$$

and bound constraints $0 \leq x_i$ for $i = 1, \dots, 5$. Constraint g_1 as well as four of the bound constraints are active at the optimum, which has an objective function value of -5000 . Point $\mathbf{x} = (250, 250, 250, 250, 250)^T$ serves as the starting point.

Problem 2.41: (Schwefel [19]) Minimise

$$f(\mathbf{x}) = -\sum_{i=1}^5 ix_i$$

subject to

$$g_1(\mathbf{x}) = \sum_{i=1}^5 (9 + i)x_i - 50000 \leq 0$$

and bound constraints $0 \leq x_i$ for $i = 1, \dots, 5$. Constraint g_1 as well as four of the bound constraints are active at the optimum, which has an objective function value of $-125000/7$. Point $\mathbf{x} = (250, 250, 250, 250, 250)^T$ serves as the starting point.

Problem HB: (Himmelblau [8]) Minimise

$$f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 \\ + 37.293239x_1 - 40792.141$$

subject to

$$g_1(\mathbf{x}) = -h_1(\mathbf{x}) \leq 0 \\ g_2(\mathbf{x}) = h_1(\mathbf{x}) - 92 \leq 0 \\ g_3(\mathbf{x}) = 90 - h_2(\mathbf{x}) \leq 0 \\ g_4(\mathbf{x}) = h_2(\mathbf{x}) - 110 \leq 0 \\ g_5(\mathbf{x}) = 20 - h_3(\mathbf{x}) \leq 0 \\ g_6(\mathbf{x}) = h_3(\mathbf{x}) - 25 \leq 0$$

where

$$h_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 \\ + 0.00026x_1x_4 - 0.0022053x_3x_5 \\ h_2(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 \\ + 0.0029955x_1x_2 + 0.0021813x_3^2 \\ h_3(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 \\ + 0.0012547x_1x_3 + 0.0019085x_3x_4$$

and bound constraints $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, and $27 \leq x_i \leq 45$ for $i = 3, 4, 5$. One of the constraints other than the bound constraints as well as three of the bound constraints are active at the optimum, which has an objective function value of -30665.539 .