



HAL
open science

Augmented Lagrangian, penalty techniques and surrogate modeling for constrained optimization with CMA-ES

Paul Dufossé, Nikolaus Hansen

► **To cite this version:**

Paul Dufossé, Nikolaus Hansen. Augmented Lagrangian, penalty techniques and surrogate modeling for constrained optimization with CMA-ES. GECCO 2021 - The Genetic and Evolutionary Computation Conference, Jul 2021, Lille / Virtual, France. 10.1145/3449639.3459340 . hal-03196365v2

HAL Id: hal-03196365

<https://inria.hal.science/hal-03196365v2>

Submitted on 25 May 2021

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.

Augmented Lagrangian, penalty techniques and surrogate modeling for constrained optimization with CMA-ES

Paul Dufossé
Thales DMS
Inria, CMAP, IP Paris
Paris area, France

Nikolaus Hansen
Inria
CMAP, CNRS, Ecole Polytechnique, IP Paris
91128 Palaiseau, France

ABSTRACT

In this paper, we investigate a non-elitist Evolution Strategy designed to handle black-box constraints by an adaptive Augmented Lagrangian penalty approach, AL- $(\mu/\mu_w, \lambda)$ -CMA-ES, on problems with up to 28 constraints. Based on stability and performance observations, we propose an improved default parameter setting. We exhibit failure cases of the Augmented Lagrangian technique and show how surrogate modeling of the constraints can overcome some difficulties. Several variants of AL-CMA-ES are compared on a set of nonlinear constrained problems from the literature. Simple adaptive penalty techniques serve as a baseline for comparison.

CCS CONCEPTS

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

KEYWORDS

Constrained optimization, Penalty Techniques, Evolution Strategies, CMA-ES, Augmented Lagrangian

ACM Reference Format:

Paul Dufossé and Nikolaus Hansen. 2021. Augmented Lagrangian, penalty techniques and surrogate modeling for constrained optimization with CMA-ES. In *2021 Genetic and Evolutionary Computation Conference (GECCO'21)*, July 10–14, 2021, Lille, France. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3449639.3459340>

1 INTRODUCTION

Evolutionary algorithms (EAs) are optimization algorithms originally introduced to solve unconstrained black-box optimization problems. In real-valued search spaces, Covariance Matrix Adaptation Evolution Strategies (CMA-ES) [10, 11] were shown to perform well in many scenarios [12]. Several constraint handling techniques have been proposed to extend CMA-ES to constrained black-box optimization. We can distinguish two main ways to handle constraints in an ES: in the search space, biasing the sample distribution toward feasible solutions, in particular by resampling or repairing infeasible solutions [2, 22, 23]; or at the fitness level, when evaluating the candidate points, such that the fitness reflects both, objective minimization and constraint satisfaction [3, 21].

This paper focuses on the second type of methods to handle constraints by adapting an Augmented Lagrangian (AL) fitness

function. This idea has been adopted in a broad sense in the optimization literature, where a constrained optimization problem is often transformed into a sequence of unconstrained optimization subproblems. With an ES as the underlying unconstrained solver, this adaptation happens at each generation and the subproblems translate into a fitness function that depends on the current state of the optimization process.

Arnold and Porter [3] discuss successful usages of the AL fitness with EAs and analyse the (1+1)-ES with AL on a sphere function with a single linear constraint, which allows to derive in-the-loop update rules of the penalty coefficients to achieve desirable performance on functions with different scales.

This update scheme has been followed by Atamna et al. [4] who applied for the first time AL to the $(\mu/\mu_w, \lambda)$ -CMA-ES with Median Success Rule [1], and assess its performance on a set of convex quadratic and ill-conditioned functions with a single linear constraint. Later, Atamna et al. [5] extend the algorithm to handle many constraints and study the case of an AL fitness with a step-size adaptive $(\mu/\mu_w, \lambda)$ -ES on sphere and ellipsoid objective functions under multiple active linear constraints. They establish linear convergence properties under sufficient stability conditions for the underlying Markov Chain, and provide numerical results.

In this paper, we study different AL approaches to handle multiple, possibly inactive inequality constraints with CMA-ES based on the existing work for Evolution Strategies [3–5]. In Section 2, we state the AL fitness function and a class of adaptive penalty techniques. In Section 3, we describe the AL update equations, investigate two variants of the AL fitness and different parameter settings in numerical experiments in combination with $(\mu/\mu_w, \lambda)$ -CMA-ES. In Section 4, we describe situations where this algorithm fails due to the shape of the constraint function at the boundary, and propose a way to overcome these difficulties with a surrogate linear model for the constraint. The different algorithms are compared on a set of benchmark problems from the EA literature in Section 5.

Notations

We introduce notations that will be used consistently throughout this paper. We denote by \mathbf{x} a column vector, \mathbf{x}^\top its transpose and x_i its i -th coordinate. The zero vector in dimension n is denoted by $\mathbf{0}_n$. Thus $\mathbf{x}^\top \mathbf{y}$ is the dot product. An interval of real numbers is written $[a, b]$ and an integer interval $[a \dots b]$. The indicator function is $\mathbb{1}_{\{A\}}$ and returns 1 if condition A is true, 0 otherwise.

GECCO '21, July 10–14, 2021, Lille, France

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *2021 Genetic and Evolutionary Computation Conference (GECCO'21)*, July 10–14, 2021, Lille, France, <https://doi.org/10.1145/3449639.3459340>.

2 FITNESS FUNCTIONS FOR CONSTRAINED OPTIMIZATION

We consider the constrained continuous search problem consisting of the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and m' generally non-linear inequality constraints $g_k : \mathbb{R}^n \rightarrow \mathbb{R}$, $k = 1, \dots, m'$:

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && f(\mathbf{x}) \\ & \text{s.t.} && g_k(\mathbf{x}) \leq 0 \quad k = 1, \dots, m' \\ & && \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned} \quad (1)$$

where $\mathbf{l}, \mathbf{u} \in [-\infty, +\infty]^n$ and we consider finite bounds as ordinary constraints, hence the optimization problem in \mathbb{R}^n has overall m constraints, $m' \leq m \leq m' + 2n$. A point \mathbf{x} is called *feasible* if it satisfies all constraints. The minimum objective value in the feasible domain is denoted as $f_c^* := f(\mathbf{x}^*)$ where $\mathbf{x}^* \in \mathbb{R}^n$ is a global feasible minimum. A constraint g_k is called *active* at \mathbf{x} if $g_k(\mathbf{x}) = 0$.

The problem definition covers any equality constraint $h(\cdot)$ with the transformation $g_h(\mathbf{x}) = |h(\mathbf{x})| - \epsilon$ where $\epsilon \geq 0$ is the desired numerical tolerance or by expressing the equality constraint with two inequality constraints.

2.1 Augmented Lagrangian fitness

The AL fitness function for problems with inequality constraints can be seen as a proximal point approach of the Lagrange dual function [19, p. 523f]. Given f and g_k , from above, we define the AL fitness $H(\cdot)$ as:

$$H(\mathbf{x}, \boldsymbol{\gamma}, \boldsymbol{\omega}) = f(\mathbf{x}) + \sum_{k=1}^m \begin{cases} \gamma_k g_k(\mathbf{x}) + \frac{\omega_k}{2} g_k^2(\mathbf{x}) & \text{if } \gamma_k + \omega_k g_k(\mathbf{x}) \geq 0 \\ -\frac{\gamma_k^2}{2\omega_k} & \text{otherwise} \end{cases} \quad (2)$$

where $\boldsymbol{\gamma}, \boldsymbol{\omega} \in \mathbb{R}^m$ are the, generally positive, Lagrangian and penalty coefficients. Figure 1 shows the AL fitness H for the sphere function with a linear constraint $x_1 \geq 1$ and three different values for γ_1 .

Necessary conditions for $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ under which the global optimum \mathbf{x}^* of a differentiable constrained problem is a local optimum of $H(\cdot, \boldsymbol{\gamma}, \boldsymbol{\omega})$ are (similar to the KKT conditions)

$$-\nabla f(\mathbf{x}^*) = \sum_{k \in \mathcal{A}^*} \gamma_k \nabla g_k(\mathbf{x}^*) \quad \text{and} \quad \forall k \notin \mathcal{A}^*: \frac{\gamma_k}{\omega_k} < \underbrace{-g_k(\mathbf{x}^*)}_{>0} \quad (3)$$

where $\mathcal{A}^* := \mathcal{A}(\mathbf{x}^*)$ is the set of indices of the constraints that are active at the optimum. If $k \in \mathcal{A}^*$, γ_k^* is the associated Lagrange multiplier. Otherwise, the complementary slackness condition requires $\gamma_k^* = 0$ but the condition $\gamma_k^* + \omega_k g_k(\mathbf{x}) < 0$ is sufficient for the AL fitness. The optimum of the constrained problem \mathbf{x}^* is the global optimum of $H(\cdot, \boldsymbol{\gamma}^*, \boldsymbol{\omega})$ if $\boldsymbol{\omega}$ is large enough.

If $\boldsymbol{\omega}$ is chosen small enough, we expect the AL fitness function generally to be well-conditioned. Sufficient conditions to obtain linear convergence have been established for a step-size adaptive ES with AL on a class of problems including convex quadratic functions with linear constraints [5].

An adaptive AL algorithm should converge to the correct Lagrangian coefficients $\boldsymbol{\gamma}$ which in effect determine the location of the optimum.

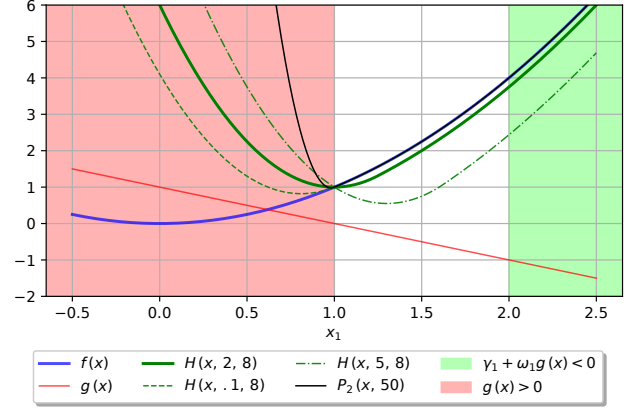


Figure 1: Illustration of the AL and quadratic penalty fitness functions. One-dimensional cut along the constrained coordinate for $f : \mathbf{x} \mapsto \|\mathbf{x}\|_2^2$ (blue) with the linear constraint $1 - x_1 \leq 0$ (red). The resulting AL fitness H is plotted with different values for the Lagrangian coefficient (estimate of the Lagrange Multiplier) in green. The minimizer of $\mathbf{x} \mapsto H(\mathbf{x}, 2, \omega)$ is the optimal solution of the constrained problem for any $\omega \geq 0$. The quadratic penalty fitness is shown with penalty coefficient $c = 50$ in black.

2.2 Penalty techniques

Penalty techniques have been widely used within evolutionary algorithms for continuous constrained problems [7, 18] and in applications [8]. In this paper, we apply two simple penalty techniques for baseline comparison. To the best of our knowledge, there exists no performance data of CMA-ES on problems with nonlinear constraints when using such penalties. The general penalty fitness reads:

$$P_\alpha(\mathbf{x}, \mathbf{c}^{(t)}) = f(\mathbf{x}) + \sum_{k=1}^m c_k^{(t)} [g_k^+(\mathbf{x})]^\alpha \quad (4)$$

where $g_k^+(\mathbf{x}) = \max[0, g_k(\mathbf{x})]$. We call P_1 the *linear* and P_2 the *quadratic* penalty fitness. The quadratic penalty fitness is a special case of the AL fitness as $P_2(\cdot, \mathbf{c}) = H(\cdot, \mathbf{0}, 2\mathbf{c})$ and is illustrated in Figure 1. The linear penalty fitness is not smooth but *exact*: if, for each active constraint, the penalty coefficient is not smaller than the Lagrange multiplier, then the minimum of the fitness is the minimum of the constrained problem. In contrast, the quadratic penalty fitness is smooth but the penalty coefficient of an active constraint must diverge to allow convergence of the optimum of P_2 to a feasible point, see also [19, p. 502ff]. Such techniques are likely to create highly ill-conditioned problems, however, CMA-ES is designed to handle ill-conditioned problems well [9].

In both cases, the penalty vector $\mathbf{c}^{(t)}$ is adapted along the optimization process as follows: when the incumbent solution (the mean of the search distribution) is infeasible in the k -th constraint, i.e. $g_k(\mathbf{m}^{(t+1)}) > 0$, we multiply the associated penalty factor by a constant parameter $c_k^{(t+1)} \leftarrow \chi_c c_k^{(t)}$. We set $\chi_c = 2^{1/\sqrt{n}}$.

3 INVESTIGATING AL PARAMETERS

A variant of an AL Evolution Strategy has been described in [3] and adapted to the non-elitist scenario in [4, 5] introducing damping factors d_γ and d_ω . Here, we look at these algorithms under a common framework (redefining $\chi := \chi^{1/d_\omega}$ if needed) and give the equivalent parameter settings from the different papers according to our parametrization. The AL update is described in Section 3.1.

We also investigate static parameter settings and provide some hints on the effect of χ in Section 3.3 and k_1 in Section 3.2 on linearly box-constrained quadratic problems. The results are compared with previous proposals [3–5].

3.1 Augmented Lagrangian coefficients update

Methods 1 show the adaptation of the coefficients γ (Lagrangian) and ω (penalty) of the AL fitness in Equation (2). The update of the AL coefficients, applied at each iteration, is based on the constraint value at the ES incumbent solution $\mathbf{m}^{(t+1)}$ which, in our case, is the mean estimator. The Lagrangian coefficient update of the k -th constraint follows $\gamma_k^{(t+1)} = \max[0, \gamma_k^{(t)} + \frac{\omega_k^{(t)}}{d_\gamma} g_k(\mathbf{m}^{(t+1)})]$.

The damping factor $d_\gamma = 5$ was introduced in [5] and appears to change the performance (number of function calls) on the test functions considered in Section 5 at most by a factor 2. In this work, we use $d_\gamma = 5$ for all variants of AL.

Line 6 in Methods 1 prevents the update of the penalty coefficients if the constraint value is clearly feasible and does not affect the AL fitness landscape (see Figure 1, green area). Updating the penalty coefficient of an inactive constraint may have unpredictable effects if the constraint becomes violated again later in the search process. The Lagrangian coefficient is updated before we test for this condition. Hence, if $g_k(\mathbf{m}^{(t+1)}) < 0$, the update implies that $\gamma_k^{(t)}$ becomes zero as the number of iterations t increases. This is compliant with the complementary slackness condition.

We also consider the variant of AL with a single penalty coefficient mentioned in [3, 19]. The AL fitness with a single penalty coefficient is the same as in Equation (2) and we have $\omega_k^{(t)} = \omega_1^{(t)}$ for all $k = 1, \dots, m$. The coefficient $\omega_1^{(t)}$ is increased if at least one condition of the update procedure in Methods 1 is fulfilled for at least one constraint. The single penalty coefficient can be seen as an upper bound of the individual penalty coefficients.

3.2 Controlling the condition number

In preliminary experiments, we observed a slowdown of AL-CMA-ES on some problems with the original default setting $k_1 = 3$ and relatively low initial penalty coefficients. The instability effect has already been remarked in [3]: “however, [this setting] is unstable and not sufficient to ensure linear convergence under all conditions tested. The rule breaks down and leads to premature stagnation when values of [here: ω] decrease too fast”. It is possible to mitigate this effect by setting the initial penalty vector $\omega^{(0)}$ appropriately. However, it does not guarantee that the algorithm is stable.

The default setting $k_1 = 3$ is justified in [3] as to balance the search minimizing the objective function and satisfying the constraint penalty. Increasing this parameter usually implies a higher condition number of the constructed fitness function given to the

Methods 1 AL initialization and update with m penalty factors

```

0 procedure init_al():           initialize the AL fitness
1    $\chi = 2^{1/\sqrt{2n}}$ ,  $k_1 = 10$ ,  $k_2 = 5$ ,  $d_\gamma = 5$ 
2    $\gamma^{(0)} \in \mathbb{R}^m$ ,  $\omega^{(0)} \in \mathbb{R}^{+,m}$ 
3 procedure update_al(g):       given the new mean vector  $\mathbf{m}^{(t+1)}$ 
4   for  $k = 1, \dots, m$ :
5      $\gamma_k^{t+1} = \max[0, \gamma_k^{(t)} + \frac{\omega_k^{(t)}}{d_\gamma} g_k(\mathbf{m}^{(t+1)})]$ 
6     if  $g_k(\mathbf{m}^{(t+1)}) > -\frac{\gamma_k^{(t)}}{\omega_k^{(t)}}$ :
7        $\omega_k^{t+1} = \begin{cases} \omega_k^{(t)} \chi^{1/4} & \text{if } \omega_k^{(t)} g_k^2(\mathbf{m}^{(t+1)}) < \\ & k_1 \frac{|H(\mathbf{m}^{(t+1)}, \gamma^{(t)}, \omega^{(t)}) - H(\mathbf{m}^{(t)}, \gamma^{(t)}, \omega^{(t)})|}{n} \\ & \text{or} \\ & k_2 |g_k(\mathbf{m}^{(t+1)}) - g_k(\mathbf{m}^{(t)})| < |g_k(\mathbf{m}^{(t)})| \\ \omega_k^{(t)} \chi^{-1} & \text{otherwise} \end{cases}$ 

```

ES. An increased condition number by one or two orders of magnitude can be handled easily by CMA and is expected to reduce the chance to diverge to the infeasible domain (as we allow the penalty term to be greater than the objective term).

We consider the following constrained problem with a convex quadratic objective function and axis-parallel linear constraints which are all active at the constrained optimum:

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && (\mathbf{x} - \mathbf{e})^\top C (\mathbf{x} - \mathbf{e}) \\ & \text{s.t.} && x_k \leq 0 \quad k = 1, \dots, m \quad (m \leq n) \end{aligned} \quad (5)$$

where $e_k = 1$ if $k \leq m$ and 0 otherwise. The unconstrained minimum is then \mathbf{e} and the constrained minimum is $\mathbf{0}_n$ such that $f_c^* = m$.

Figure 2 displays the results for several runs of AL-CMA-ES with varying parameter k_1 on a sphere and an ellipsoid objective (where the condition number is 10^6) with different number of constraints. On the constrained sphere problems, a higher value for $k_1 = 10$ or 20 also shows improved convergence speed, i.e. the number of function evaluations needed to approach the optimum value f_c^* can be reduced roughly by a factor 1.5 at the price of an increase in the condition number that also depends on the number of constraints. The case $m = 10$ is special as all coordinates are constrained.

On ellipsoid problems, there is no gain of convergence speed, but a value of $k_1 = 10$ decreases the strategy’s condition number by a factor 10 when $m \in \{2, 6\}$. In the case when $m = 10$, different runs lead to very different condition numbers, from 10 to 10^7 .

The results on the sphere function suggest that a default setting of $k_1 = 10$ can improve convergence speed in some situations. In Section 5 we illustrate this parameter tuning on a set of nonlinear problems from the literature.

3.3 The penalty coefficient update

The default parameter setting (to be used with methods 1) proposed by Arnold and Porter [3] is $d_\gamma = 1$, $\chi = 2^{1/4} \approx 1.189$, $k_1 = 3$, $k_2 = 5$. Atamna et al. chose χ as a function of the dimension $\chi = 2^{1/n}$ [4] and later add a significant damping factor [5] and lessen the penalty update: $d_\gamma = 5$, $\chi = 2^{1/5n}$, $k_1 = 3$, $k_2 = 5$ where n is the problem

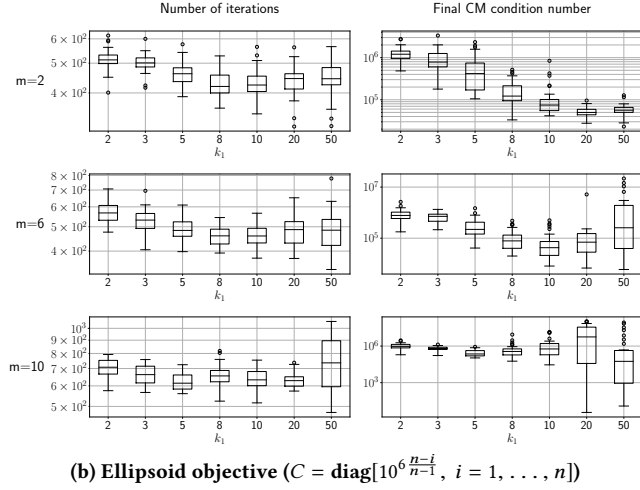
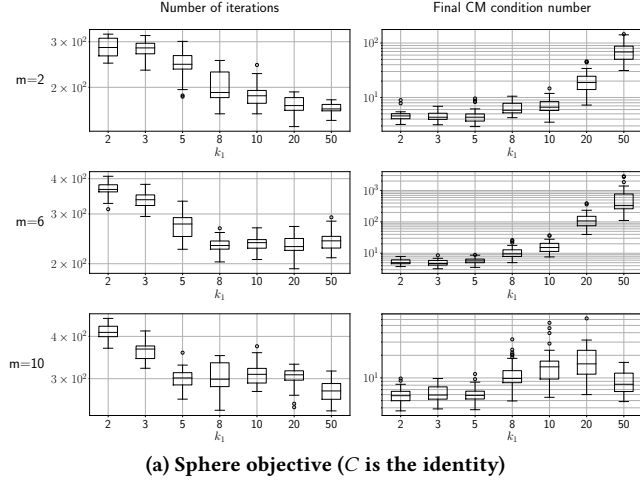


Figure 2: Boxplots showing the number of iterations needed by the ES before $|f(m^{(t)}) - f_c^*| < 10^{-5}$ (left), and final condition number of the covariance matrix in CMA-ES (right). For each value of k_1 , we see data from 50 runs of AL- $(\mu/\mu_w, \lambda)$ -CMA-ES in dimension $n = 10$ starting from the same feasible point. Other parameters are fixed: $\chi = 2^{\frac{1}{4}}$, $d_\gamma = 5$

dimension. The latter choice has been used when comparing AL-CMA-ES with other constraint handling techniques [23].

We investigate how the χ parameter of the AL- $(\mu/\mu_w, \lambda)$ -CMA-ES impacts performance when the initial Lagrangian coefficient $\gamma_1^{(0)}$ is far from the Lagrange multiplier. Consider the problem defined in Equation (5) when we minimize an ellipsoid function under a single constraint on the first coordinate, i.e. $m = 1$. We expect γ_1 to approach the value $\gamma_1^* = 2 \cdot 10^6$, which is the Lagrange multiplier of this constrained problem. Initial settings for AL coefficients are $\gamma_1^{(0)} = 0$, $\omega_1^{(0)} = 1$. We want to find a default χ value that does not result in either too slow adaptation or instability.

We illustrate this performance trade-off for problem dimension ranging from 3 to 40, and consider the case where there is only one constraint. We display χ on a log-log scale that we consider

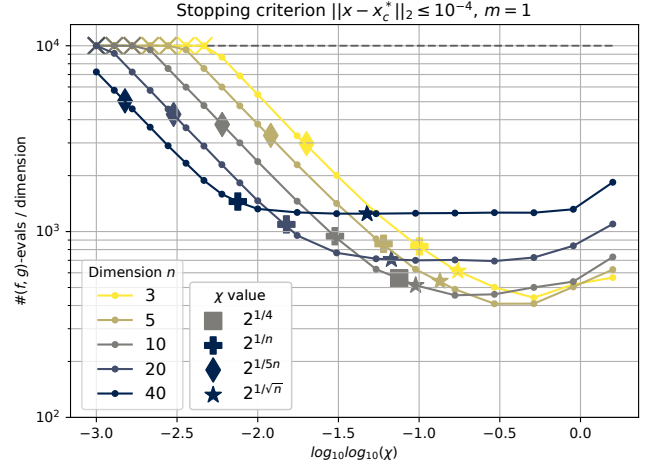


Figure 3: Mean number of function evaluations needed for the AL-CMA-ES to reach the termination condition given in the title when varying χ , one constraint, $k_1 = 10$, 10 runs averaged. Each graph shows a different search space dimension. The + signs indicate the recommended dimension-dependant χ parameter value from [4], and the diamonds refer to [5], which differs only by a damping factor. The square indicates the parameter used for dimension 10 in [3] and the stars our new recommendation. A cross at the maximum number of evaluations, 10^4 , indicates that at least one run reached the maximum number of (f, g) -evals before terminating.

to be a natural choice: let $p \in \{1/4, -1\}$ and consider the penalty parameter update: $\omega \leftarrow \omega \cdot \chi^p \xleftrightarrow{\omega > 0} \log \omega \leftarrow \log \omega + (\log \chi) \cdot p$, and since $\log \chi > 0$ we can see it as a strictly positive learning rate we want to display on the log-scale. Results can be found in Figure 3. We also display recommendations from previous papers for comparison.

For all dimensions, values below $2^{1/n}$ slow down the optimization process because it takes many iterations to adapt the Lagrangian coefficients. If $\chi \geq 10$, the algorithm takes too big steps in adapting the coefficients. In dimensions 20 and 40, there is a performance plateau. The width of the performance plateau, or the interval of reasonable values for χ , is increasing with dimension. An optimal, still stable, choice for χ is at the leftmost part of this plateau. We suggest a value of $\chi = 2^{1/\sqrt{n}}$ which will be used in the benchmark Section 5.

4 SURROGATE MODELING OF THE CONSTRAINTS

The AL technique for Evolution Strategies has been designed to handle black-box constrained continuous optimization problems with smooth objective and constraints functions at the boundary. We exhibit cases where these assumptions are violated and lead to a failure in Section 4.1 and propose a remedy in Section 4.2 by replacing the original constraints with a surrogate model used in the fitness function.

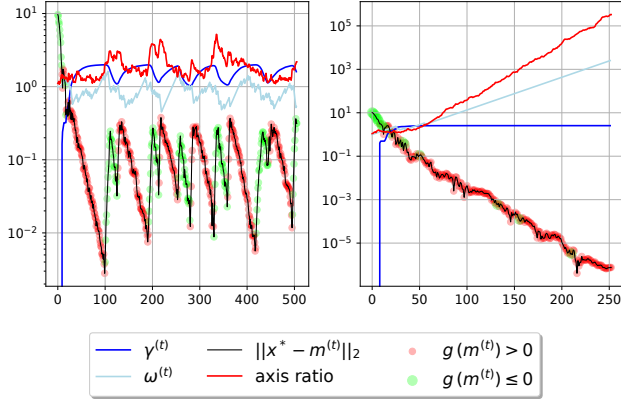


Figure 4: Some metrics of AL-CMA-ES in two cases where it does not behave as desired on a piecewise-linear constraint which is non-differentiable at the boundary. On the left: $g(x) = (1 + \mathbb{1}_{\{x_1 \leq 0\}})x_1$, the Lagrangian coefficient (in blue) oscillates between the 2 values defined by each side of the piece-wise linear constraint; on the right: $g(x) = \mathbb{1}_{\{x_1 > 0\}}x_1$, the penalty coefficient and the axis ratio of CMA-ES are exponentially increasing. The distance between $m^{(t)}$ and the optimum is colored in green if $m^{(t)}$ is feasible, red otherwise.

4.1 Limits of the Augmented Lagrangian

Consider the problem defined in Equation (5) with a sphere objective and one linear constraint $x_1 \leq 0$. From this problem, we construct a non-smooth test problem: we transform the linear constraint into a piecewise linear constraint $G(x_1) \leq 0$ where G is a sign-preserving, non-decreasing transformation. The constraint is not anymore differentiable in $x_1 = 0$. We monitor the AL coefficients for two such cases in Figure 4.

- In Figure 4, left, the function is piecewise linear and concave. The Lagrange multiplier depends on the considered halfspace (here $\gamma^* \in \{1, 2\}$) and the Lagrangian coefficient estimate oscillates between the two values, hence the AL- $(\mu/\mu_w, \lambda)$ -CMA-ES does not converge to the constrained optimum.
- In Figure 4, right, the function is constant zero in the feasible domain. Hence g is nowhere negative and the Lagrangian coefficient $\gamma^{(t)}$ can never decrease. After it reaches a value greater than the Lagrange multiplier $\gamma^* = 2$ it can never approach the optimal value again. The resulting fitness function is similar to the linear penalty, as the quadratic term in g vanishes at the boundary.

The latter scenario may happen in a real-world application when a simulation crashes after time $T(\mathbf{x})$ and we then define a constraint as $g_T(\mathbf{x}) = 1/T(\mathbf{x})$ if the simulation fails and otherwise as $g_T(\mathbf{x}) = 0$. Fitting a linear model of the constraint (with careful handling of the zero intercept, see below) then builds an extension of the constrained problem into the feasible domain.

4.2 Fitting a linear model of the constraints

In [6], the authors improve a surrogate-aided algorithm for efficient constrained optimization proposed in [20]. It relies on surrogate

modeling of both the objective and constraints using radial basis functions. Performance is assessed for up to a few hundred function evaluations.

Replacing only the constraints by a linear model has been investigated by Kramer et al. [13, 14] in the case of underlying linear constraints where the constraints function only return a boolean. The procedure to estimate the linear model is triggered only once when the first infeasible candidate is sampled.

The authors also claim that “a linear surrogate approximation of non-linear constraints is sufficient and leads to amazingly good results” [14], without more details about the algorithm or the results.

In the following, we consider two cases that are more informative than binary-valued outputs:

- **Case 1:** the usual situation where the constraints are continuous and can be evaluated everywhere. Using a surrogate model can still be useful if constraints are non-differentiable or particularly expensive to evaluate.
- **Case 2:** the constraints do not have a meaningful value in the feasible domain (hence are defined as zero for feasible solutions).

To address **Case 1**, we build a linear model, $\hat{g}_k^{(t)}$, at each time step t for constraint k using the $\kappa = 2n$ last k -infeasible points, denoted as $\mathbf{D}_k^{(t)}$, and their g_k -values. Because the value of active constraints equals zero at the optimum, we are particularly interested in a sign preserving model, that is: $g_k(\mathbf{x})\hat{g}_k^{(t)}(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbf{D}_k^{(t)}$. For this reason, we fit the linear regression without intercept on a transformed dataset. We first center the data around a point $\mathbf{c}^{k,(t)}$ assuming for the moment that $\mathbf{c}^{k,(t)} = \mathbf{m}^{(t)}$, and transform the dataset $\mathbf{D}^{(t)}$:

$$\tilde{\mathbf{x}} := \mathbf{x} - \mathbf{c}^{k,(t)}, \quad \tilde{y}(\mathbf{x}) := g_k(\mathbf{x}) - g_k(\mathbf{c}^{k,(t)}) \quad (6)$$

Then fit the linear regression with least squares to obtain:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^n} \sum_{\mathbf{x} \in \mathbf{D}^{(t)}} (\boldsymbol{\theta}^\top \tilde{\mathbf{x}} - \tilde{y}(\mathbf{x}))^2 \quad (7)$$

Finally we define the constraint surrogate as:

$$\hat{g}_k^{(t)}(\mathbf{x}) := \hat{\boldsymbol{\theta}}^\top (\mathbf{x} - \mathbf{c}^{k,(t)}) + g_k(\mathbf{c}^{k,(t)}) \quad (8)$$

This procedure guarantees that the model is exact at $\mathbf{c}^{k,(t)}$, i.e. $\hat{g}_k^{(t)}(\mathbf{c}^{k,(t)}) = g_k(\mathbf{c}^{k,(t)})$. If the constraint g_k is active in \mathbf{x}^* we expect $g_k(\mathbf{m}^{(t)}) \xrightarrow[t \rightarrow \infty]{} 0$, and hence in the limit the linear model $\hat{g}_k^{(t)}$ is exact at the location of the optimum. This algorithm is described in Method 2.

In the scenario of **Case 2**, the choice of $\mathbf{c}^{k,(t)}$ is more intricate because $g_k(\mathbf{m}^{(t)}) = 0$ for any feasible $\mathbf{m}^{(t)}$. In this case, we identify $\mathbf{c}^{k,(t)}$ with a line search performing interval halving between $\mathbf{m}^{(t)}$ and its closest infeasible neighbour in $\mathbf{D}^{(t)}$. The number of iterations S of the line search is a parameter of the optimization algorithm. First experiments suggest that $S = 3$ is enough to maintain a performance comparable to the **Case 1**.

We skip the model update procedure for the k -th constraint when all candidates are feasible in this constraint *and* they are assigned negative values by the current surrogate model. Note that the new linear model is fitted with a least squares regression and it is still possible that this model assigns a feasible solution to be infeasible.

Method 2 MM-AL- $(\mu/\mu_w, \lambda)$ -CMA-ES

```

0 init_al()
1 initialize CMA-ES( $\mathbf{m}^{(0)} \in \mathbb{R}^n, \sigma^{(0)} \in \mathbb{R}_+, C^{(0)} \in \mathbb{R}^{n \times n}$ ),
  and iteration counter  $t = 0, \kappa \in \mathbb{N}$ 
2 while stopping criterion not met:
3   Ask population:
      $\mathbf{X}^{(t)} = [\mathbf{x}^{i:(t)} \sim \mathcal{N}(\mathbf{m}^{(t)}, [\sigma^{(t)}]^2 C^{(t)})]_{i=1, \dots, \lambda}$ 
4   for  $k = 1, \dots, m$ :
5     if not all ( $g_k(\mathbf{x}) \leq 0$  and  $\hat{g}_k^{(t)}(\mathbf{x}) \leq 0; \mathbf{x} \in \mathbf{X}^{(t)}$ ):
6       update the linear surrogate  $\hat{g}_k^{(t)}$  with  $D_k^{(t)}$ 
7   Tell CMA-ES with fitness  $H(\cdot, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\omega}^{(t)}, \hat{g}^{(t)})$ 
8   update_al( $\hat{g}^{(t)}$ ) using  $\mathbf{m}^{(t+1)}$ 
9    $t \leftarrow t + 1$ 

```

The algorithm combining AL- $(\mu/\mu_w, \lambda)$ -CMA-ES with a surrogate model of the constraints is described in Method 2. We slightly abuse our notation where $H(\cdot, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\omega}^{(t)}, \hat{g}^{(t)})$ denotes the original AL fitness function described in Equation (2) but instead of the original constraints functions we pass the surrogate model of the constraints. The same is done for the procedure update_al() in Methods 1. The question how these changes affect the behaviour of AL-CMA-ES on smooth non-linear constrained problems is considered in the next section.

5 EVALUATION ON NONLINEAR PROBLEMS

We compare the active (1+1)-CMA-ES [2] and the $(\mu/\mu_w, \lambda)$ -CMA-ES with either the quadratic or linear penalty technique described in Section 2.2 or a variant of an AL fitness function on the original problem in **Case 1**, and 2 variants of AL (with or without a metamodel) on the restricted problem in **Case 2**.

The set of linear and nonlinear constrained problems is taken from [2]. These problems also appear in the CEC 2006 competition [16] and in [14, 15]. More details about the problems implementation can be found in Appendix A.

All the problems are unimodal, they have dimension ranging from 2 to 10 and 1 to 8 inequality constraints. Problems G6, G7, G9, G10 and HB have lower and upper bounds constraints, Problems 2.40 and 2.41 have only lower bounds and Problem TR2 has no bound constraint. The bound constraints are converted as general (linear) inequality constraints by the AL. Table 1 gives a summary of the problems characteristics.

When mentioned, we use the starting point recommended for the test problem (i.e. for 2.40, 2.41 and TR2). Otherwise, we start with a point that is randomly picked in a dataset of feasible points generated before we start the benchmarking process. To generate feasible points, we run CMA-ES minimizing $\sum_{k=1}^m g_k^+(\mathbf{x})$ so that any point in the feasible domain is optimal with 0 as fitness value. We stop after 1,000 feasible data points are found and repeat this for 10 runs, resulting in a dataset of 10,000 points for initialization. Note that only the (1+1)-CMA-ES of [2] requires a feasible point to start with, whereas the AL approach does not.

Algorithms are stopped when a feasible candidate solution \mathbf{x} satisfies the condition $|f(\mathbf{x}) - f_c^*| \leq 10^{-8} |f_c^*|$ for the first time. We

problem	n	m'	m	m_{act}	m_{lin}	Starting point
G6	2	2	6	2	0	Random
G7	10	8	28	6	3	Random
G9	7	4	18	2	0	Random
G10	8	6	22	6	3	Random
TR2	2	1	1	1	1	Fixed
2.40	5	1	6	5	1	Fixed
2.41	5	1	6	5	1	Fixed
HB	5	3	13	4	0	Random

Table 1: Characteristics for each problem in the testbed: dimension n , number of non-bound constraints m' and total number of constraints m , number of active constraints at the optimum m_{act} and number of linear constraints among the m' general constraints.

stop the algorithm if the best found objective value of a feasible candidate has not improved in the last 2,000 function evaluations.

5.1 Algorithmic setup

The generic inequality-constrained black-box optimization problem 1 is transformed into an unconstrained problem, either through the adaptive AL fitness or a quadratic penalty, and solved with the $(\mu/\mu_w, \lambda)$ -CMA-ES. The CMA-ES uses Cumulative Step-size Adaptation (CSA, as opposed to the Median Success Rule step-size adaptation previously used in [4]) and negative weights for the Covariance Matrix update [10]. We run all experiments using the pycma package [11] with the default parameter setting for CMA-ES are unchanged.

Initialization heuristics. For CMA-ES, the initial step-size is always $\sigma^{(0)} = 1$. The initial covariance matrix $C^{(0)}$ is diagonal and the variance in each coordinate is determined by the bound constraints: $C^{(0)} = \text{diag}([\frac{u_i - l_i}{5}, i = 1, \dots, n])^2$. For unbounded Problems TR2, 2.40, 2.41, the initial covariance matrix is set to the identity.

In a black-box scenario, the initialization of fitness coefficients associated to each constraint depend on the initial population. Only after a few functions evaluations one can fine-tune the initial AL coefficients from observed function values and dispersion. The initial Lagrangian coefficients are set to $\boldsymbol{\gamma}^{(0)} = \mathbf{0}$ and we propose a simple heuristic for the initial setting of the penalty coefficient $\boldsymbol{\omega}$. When the first population of λ points is sampled and evaluated, we compute $\Delta f = IDR(f(\mathbf{x}^{1,(0)}), \dots, f(\mathbf{x}^{\lambda,(0)}))$ and $\Delta g_k^2 = IDR(g_k(\mathbf{x}^{1,(0)})^2, \dots, g_k(\mathbf{x}^{\lambda,(0)})^2)$ where IDR is the inter-decile range, asserting that $\Delta g_k^2 \neq 0$ for all k (it is valid for all considered test problems and also used for **Case 2** to provide similar test conditions). We set the initial penalty coefficient of the k -th constraint to be $\omega_k^{(0)} = 10^2 \cdot \Delta f / \Delta g_k^2$. In the case of a single penalty coefficient it is straightforward to chose $\omega_1^{(0)} := \max[\omega_k^{(0)}, k = 1, \dots, m]$.

For the linear and quadratic penalty, the initial penalty vector follows the same idea with $c_k^{(0)} = 10^3 \cdot \Delta f / \Delta g_k^2$. The setting $\chi_c = 2^{1/\sqrt{n}}$ is based on cursory experimentation and aims to make the adaptation speed comparable to AL approaches.

The ten algorithms we compare are the **Linear Penalty** and **Quadratic Penalty**, the **(1+1) Active CMA-ES** and seven variants of AL techniques: **AL many** with our recommended setting ($k_1 = 10$, $\chi = 2^{1/\sqrt{n}}$), **AL many, old settings** with the previously proposed parameter setting ($k_1 = 3$, $\chi = 2^{1/(5n)}$) and many penalty coefficients, **AL single** and **AL single, old settings** with a single penalty coefficient for all constraints. **MM-AL many** replaces the constraint with a linear surrogate model and operates on the original problem (**Case 1**). **MM-AL many on Restricted CP** and **AL many on Restricted CP** operate on the restricted constrained problem (**Case 2**). When constraints values are not defined in the feasible domain, **MM-AL many on Restricted CP** applies a line search ($S = 3$ iterations) when the mean is feasible. Model building for both MM-AL-CMA-ES is only applied to the m (possibly) non-linear constraints g and not to the linear bound constraints.

5.2 Results

Figure 5 displays the Empirical Cumulative Distribution Function (ECDF) of f - or g -evaluations for the 8 test problems, where the single target is the stopping criterion defined before. The data aggregated over all problems divided by each problem dimension are shown in the lower right. The active (1+1)-CMA-ES [2] and the surrogate-based algorithm with line search need more constraints than objective function evaluations, hence we display the numbers separately. The penalty techniques evaluate the mean only for the constraints and we display only the number of f -evaluations; the g -evaluations are the same multiplied by a factor $1 + 1/\lambda$.

For the active (1+1)-CMA-ES, only the 10th, 50th and 90th number of evaluations are reported. We extrapolate its ECDF by fitting a model based on the Log-normal distribution. This model is exact at the data points. The estimation technique is detailed in Appendix B.

We conducted Wilcoxon-Mann-Whitney tests for statistical significance to compare runtimes data for the quadratic penalty and AL techniques. They can be found in Table 2 of Appendix C.

The number of f -evaluations needed by any variant of AL-CMA-ES is greater than the number of f -evaluations needed by the **(1+1) Active CMA-ES** while the AL techniques tend to require fewer constraint evaluations. Only on problem TR2, no AL variant or the penalty technique is able to improve over the number of g -evaluations needed by the active elitist strategy.

The **AL many** is up to 3 times faster than the **AL many, old settings** on problem G6. It is also significantly better on problems 2.40, 2.41. For problem G10, **AL many, old settings** fails to locate the optimum in 18 runs out of 50, whereas **AL many** failed only once. The two variants perform similarly on other problems. The observations support our recommendation for a new default parameter setting. The problem G10 also reveals a serious defect of the **AL single** which fails to converge in about 90% of the cases. With the old settings it is stable but needs more than 20,000 functions evaluations, as much as the **Linear Penalty** or the **AL many on Restricted CP**.

AL many requires 1.5 to 2 times fewer function evaluations than **AL single** on problems 2.40, 2.41 and G7. It is also faster on all problems but TR2.

All runs of **MM-AL many** are successful on all problems but G9 where a handful of outlier runs are 3 to 10 times slower than the median.

On the restricted problem of **Case 2**, **MM-AL many on Restricted CP** is 1.5 times slower in f -evaluations and more than 2 times slower in g -evaluations than **AL many on Restricted CP** on problem G6 but **MM-AL many on Restricted CP** needs two times fewer f -evaluations and as many g -evaluations as **AL many on Restricted CP** on problem G10.

Interestingly, the **AL many on Restricted CP** solves the G6 problem with a median number of function evaluations between 700 and 800, so does the **Quadratic Penalty**, whereas the other AL variants have access to more information but need around 1,000 function evaluations, which is also the number of constraint evaluations of the **(1+1) Active CMA-ES**.

The **Quadratic Penalty** is better than **AL many** on the 2-dimensional problems G6 and TR2, whereas the converse is true on the highest-dimensional problems G7 and G10 (except for $\sim 10\%$ of outliers).

Overall, **Quadratic Penalty** and **AL Many** perform best (lower right figure), where **AL Many** is ahead on problems which require a high number of function evaluations, and the **Quadratic Penalty** is slightly faster on problems which require a lower number of evaluations.

6 SUMMARY AND CONCLUSION

Starting from recent work, we empirically compared several Augmented Lagrangian fitness functions while using them with CMA-ES, handling equality and inequality and possibly nonactive constraints. This is the first work to combine Augmented Lagrangian and CMA-ES with CSA, the standard mechanism for step-size adaptation. We investigated hyperparameters related to adaptation and stability of the AL fitness in the light of performance on simple problems and also assessed these choices on non linear constrained test problems from the literature. The previous choice for static parameters does not allow a fast-enough increase of the penalty coefficients and can lead to failure in some cases. We proposed a new parameter setting for the many-coefficients variant of AL- $(\mu/\mu_w, \lambda)$ -CMA-ES, which appears as the best choice for the more difficult test problems considered. The test problems are all unimodal and the results reveal that an elitist strategy with active CMA takes roughly two times fewer objective function evaluations and two times more constraints evaluations than AL- $(\mu/\mu_w, \lambda)$ -CMA-ES. The performance of the AL- $(\mu/\mu_w, \lambda)$ -CMA-ES on multimodal test problems has not been investigated here, however on unconstrained multimodal problems $(\mu/\mu_w, \lambda)$ -CMA-ES vastly outperforms the (1+1)-CMA-ES [12].

This work also revisits the idea of a surrogate-assisted fitness function for constrained optimization. If the constraints are defined everywhere, our results suggest that a local linear approximation of nonlinear constraints can replace the original constraints while only moderately increasing the number of required evaluations. If the constraints are non-differentiable at the boundary, a surrogate model allows to build and optimize a smooth AL fitness function. If the constraints function is not defined in the feasible domain, we find cases where the surrogate-assisted version outperforms the

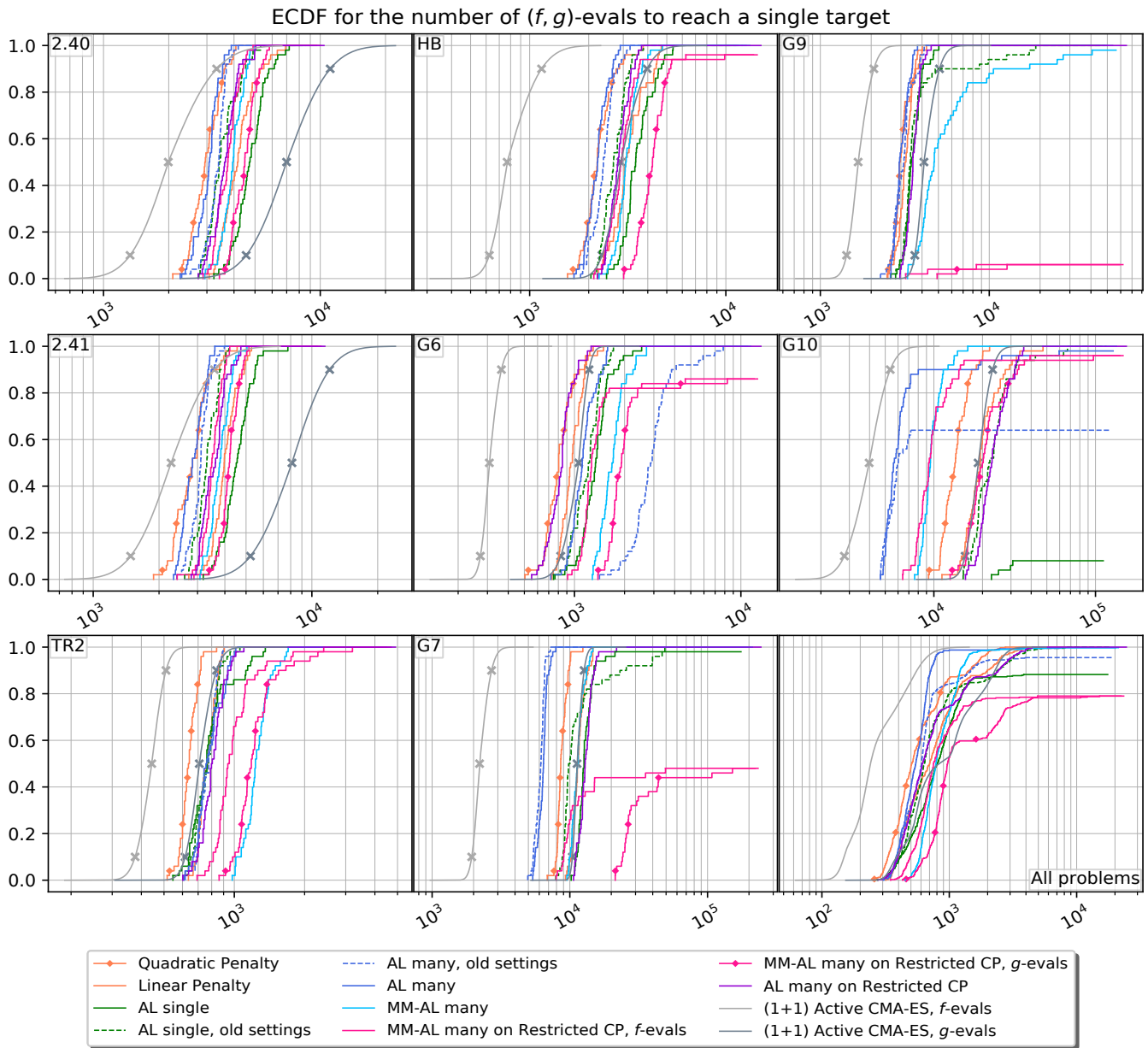


Figure 5: Single-target ECDFs of f - or g -evaluations of 50 runs on the benchmark test set. For the MM-AL many on Restricted CP and the (1+1) Active CMA-ES we display separately the number of objective and constraints functions evaluations. For other algorithms these are the same number. For the (1+1) Active CMA-ES we display the three known percentiles (crosses) as well as the fitted log-normal model. The left column is the set of problem with linear constraints 2.40, 2.41 and TR2. The lower right shows the aggregated data over all problems divided by problem dimension

genuine AL, yet it is not reliable on all test functions investigated. We plan to further exploit the surrogate model to save constraints evaluations either by surrogate evaluation or injecting repaired solutions. If we also build a quadratic surrogate model of the objective function, the resulting subproblem can be solved by sequential quadratic programming.

On our small testbed, even a simple quadratic penalty approach performed reasonably well. This observation provokes the question whether these commonly used standard test functions are indeed challenging enough, and suggests further investigation with other test functions [17, 24], in particular in higher dimensions.

REFERENCES

- [1] Ouassim Ait Elhara, Anne Auger, and Nikolaus Hansen. 2013. A Median Success Rule for Non-Elitist Evolution Strategies: Study of Feasibility. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '13)*, Blum et al. Christian (Ed.). ACM, ACM Press, Amsterdam, Netherlands, 415–422. <https://hal.inria.fr/hal-00801414>
- [2] Dirk V. Arnold and Nikolaus Hansen. 2012. A (1+1)-CMA-ES for Constrained Optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '12)*, Terence Soule and Jason H. Moore (Eds.). ACM, ACM Press, Philadelphia, United States, 297–304. <https://doi.org/10.1145/2330163.2330207>
- [3] D. V. Arnold and J. Porter. 2015. Towards an Augmented Lagrangian Constraint Handling Approach for the (1 + 1)-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '15)*. ACM Press, Madrid, Spain, 249–256.
- [4] Asma Atamna, Anne Auger, and Nikolaus Hansen. 2016. Augmented Lagrangian Constraint Handling for CMA-ES – Case of a Single Linear Constraint. In *Parallel Problem Solving from Nature – PPSN XIV*, Julia Handl, Emma Hart, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter (Eds.). Springer International Publishing, Cham, 181–191.
- [5] Asma Atamna, Anne Auger, and Nikolaus Hansen. 2017. Linearly Convergent Evolution Strategies via Augmented Lagrangian Constraint Handling. In *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (Copenhagen, Denmark) (FOGA '17)*. Association for Computing Machinery, New York, NY, USA, 149–161. <https://doi.org/10.1145/3040718.3040732>
- [6] Samineh Bagheri, Wolfgang Konen, Michael Emmerich, and Thomas Bäck. 2017. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing* 61 (2017), 377 – 393. <https://doi.org/10.1016/j.asoc.2017.07.060>
- [7] Helio J. C. Barbosa, Afonso C. C. Lemonge, and Heder S. Bernardino. 2015. *A Critical Review of Adaptive Penalty Techniques in Evolutionary Computation*. Springer India, New Delhi, 1–27. https://doi.org/10.1007/978-81-322-2184-5_1
- [8] Vinicius Veloso de Melo and Giovanni Iacca. 2014. A modified Covariance Matrix Adaptation Evolution Strategy with adaptive penalty function and restart for constrained optimization. *Expert Systems with Applications* 41, 16 (2014), 7077 – 7094. <https://doi.org/10.1016/j.eswa.2014.06.032>
- [9] N. Hansen. 2006. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, J.A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea (Eds.). Springer, Berlin, Heidelberg, New York, 75–102.
- [10] Nikolaus Hansen. 2016. The CMA Evolution Strategy: A Tutorial. , 39 pages. arXiv:1604.00772 [cs.LG] <http://arxiv.org/abs/1604.00772>
- [11] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. 2019. CMA-ES/pycma on Github, version 3.0.4. Zenodo, DOI:10.5281/zenodo.2559634. <https://doi.org/10.5281/zenodo.2559634>
- [12] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošik. 2010. Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation (Portland, Oregon, USA) (GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 1689–1696. <https://doi.org/10.1145/1830761.1830790>
- [13] Oliver Kramer. 2010. A Review of Constraint-Handling Techniques for Evolution Strategies. *Applied Comp. Int. Soft Computing* 2010 (01 2010). <https://doi.org/10.1155/2010/185063>
- [14] Oliver Kramer, André Barthelmes, and Günter Rudolph. 2009. Surrogate Constraint Functions for CMA Evolution Strategies. In *KI 2009: Advances in Artificial Intelligence*, Bärbel Mertsching, Marcus Hund, and Zaheer Aziz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 169–176.
- [15] Oliver Kramer and Hans-Paul Schwefel. 2006. On three new approaches to handle constraints within evolution strategies. *Natural Computing* 5 (11 2006), 363–385. <https://doi.org/10.1007/s11047-006-0001-x>
- [16] Jing Liang, Thomas Runarsson, Efrén Mezura-Montes, M. Clerc, Ponnuthurai Suganthan, C. Coello, and Kalyan Deb. 2006. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. *Nanyang Technological University, Singapore, Tech. Rep* 41 (01 2006).
- [17] Rammohan Mallipeddi and Ponnuthurai Suganthan. 2010. *Problem definitions and evaluation criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization*. Technical Report. Nanyang Technological University, Singapore.
- [18] Zbigniew Michalewicz and Marc Schoenauer. 1996. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evol. Comput.* 4, 1 (March 1996), 1–32. <https://doi.org/10.1162/evco.1996.4.1.1>
- [19] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (second ed.). Springer, New York, NY, USA.
- [20] Rommel Regis. 2014. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization* 46, 2 (2014), 218–243. <https://doi.org/10.1080/0305215X.2013.765000>
- [21] T. P. Runarsson and Xin Yao. 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4, 3 (2000), 284–294.
- [22] Naoki Sakamoto and Youhei Akimoto. 2019. Adaptive Ranking Based Constraint Handling for Explicitly Constrained Black-Box Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (Prague, Czech Republic) (GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 700–708. <https://doi.org/10.1145/3321707.3321717>
- [23] P. Spettel, H. Beyer, and M. Hellwig. 2019. A Covariance Matrix Self-Adaptation Evolution Strategy for Optimization Under Linear Constraints. *IEEE Transactions on Evolutionary Computation* 23, 3 (2019), 514–524. <https://doi.org/10.1109/TEVC.2018.2871944>
- [24] Guohua Wu, Rammohan Mallipeddi, and Ponnuthurai Suganthan. 2016. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization*. Technical Report. Nanyang Technological University, Singapore.

A BENCHMARK PROBLEM SET

The problems considered in Section 5 are described in paper [2] and implemented in pure Python except for problems PB2.40 and PB2.41 where we used routines from the numpy library. Also note that in references [2, 15, 18] there is a replication error defining the g_1 constraint function for the Himmelblau (HB) problem: the coefficient attached to the term x_1x_4 is given as 0.00026 whereas it should be 0.006262 [16] and we remind the correct description below:

Problem HB (Himmelblau’s nonlinear optimization problem)

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 \\ & + 37.293239x_1 - 40,792.141 \\ \text{subject to} \quad & 0 \leq g_1(\mathbf{x}) \leq 92 \\ & 90 \leq g_2(\mathbf{x}) \leq 110 \\ & 20 \leq g_3(\mathbf{x}) \leq 25 \\ & 78 \leq x_1 \leq 102 \\ & 33 \leq x_2 \leq 45 \\ & 27 \leq x_i \leq 45 \quad (i = 3, 4, 5) \end{aligned}$$

$$\begin{aligned} g_1(\mathbf{x}) &= 85.334407 + 0.0056858x_2x_5 + 0.006262x_1x_4 - 0.0022053x_3x_5 \\ g_2(\mathbf{x}) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \\ g_3(\mathbf{x}) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \end{aligned}$$

B FITTING A LOG-NORMAL DISTRIBUTION TO RUNTIMES DATA

For the active (1+1)-CMA-ES the paper [2] reports the 10-th, 50-th and 90-th quantiles, $\{m_{10}, m_{50}, m_{90}\}$. We propose to fit a model of the entire ECDF which is exact in these 3 quantiles. A convenient model for the runtime of an algorithm is a positive random variable which is the product of a high number of small factors, hence the choice of the Log-normal distribution.

A random variable X follows the Log-normal distribution $X \sim \text{LogN}(\mu, \sigma^2)$ if $Y = \ln X$ follows the normal distribution $Y \sim$

$\mathcal{N}(\mu, \sigma^2)$. We are looking for the (estimated) parameters of the log-normal $\hat{\mu}, \hat{\sigma}$.

First fit the median:

$$\hat{\mu} = \ln m_{50} \tag{9}$$

Then solve in σ for the repartition function at the 10-th and 90-th quantiles:

$$\frac{1}{2} + \frac{1}{2} \text{erf}\left(\frac{\ln x - \mu}{\sigma\sqrt{2}}\right) = c \tag{10}$$

i.e. solve Equation (10) for $\mu = \hat{\mu}$ and $(x, c) \in \{(m_{10}, 0.1), (m_{90}, 0.9)\}$, where erf is the Gauss error function

The solution is:

$$\hat{\sigma}_c = \frac{1}{\sqrt{2}}(\ln x - \hat{\mu})[\text{erf}^{-1}(2c - 1)]^{-1} \tag{11}$$

where erf^{-1} is the inverse function of erf.

We obtain two values $\hat{\sigma}_{10}, \hat{\sigma}_{90}$ which are generally different because the 3 data points do not fit perfectly a log-normal distribution).

If we denote by $\Phi(x, \mu, \sigma)$ the cumulative distribution function of a log-normally distributed random variable with parameters (μ, σ) , then our final estimate for the distribution of the runtimes, given $\{m_{10}, m_{50}, m_{90}\}$, would be:

$$\mathbb{1}_{\{x \leq m_{50}\}} \Phi(x, \hat{\mu}, \hat{\sigma}_{10}) + \mathbb{1}_{\{x > m_{50}\}} \Phi(x, \hat{\mu}, \hat{\sigma}_{90}) \tag{12}$$

C STATISTICAL COMPARISON OF ALGORITHMS DATA

Section 5 provides runtime data for many algorithm variants on several problems in the form of ECDF. Here, we test observed runtime differences for statistical significance. For each problem and each pair of algorithms, we compute the p -value from the two-sided Wilcoxon-Mann-Whitney test. If an algorithm fails, its runtime is replaced with $+\infty$, such that it’s always worse than a successful run, and comparing two failing runs results in equality. We also apply a Bonferroni correction of $\beta = 8$, that is the number of test problems. If algorithm A is better than algorithm B for a given problem with a p -value lower than $0.05/\beta$, we present the item in the corresponding entry, see Table 2.

Algorithm	Quadratic Penalty	Linear Penalty	AL single	AL single old settings	AL many old settings	AL many	MM-AL many	MM-AL many Restricted CP, f -evals	MM-AL many Restricted CP, g -evals	AL many Restricted CP
Quadratic Penalty	X	HB -11.7	HB -14.5	HB -7.0			HB -12.7	HB -10.1	HB -16.5	HB -9.5
		G6 -5.1	G6 -13.3	G6 -11.2			G6 -16.7	G6 -14.0	G6 -17.1	G7 -16.7
		G7 -15.3	G7 -16.6	G7 -8.4		G6 -17.1	G7 -15.5	G7 -11.9	G7 -17.4	G9 -8.3
		G10 -10.7	G9 -7.3	G9 -6.3		2.40 -3.6	G9 -15.6	G9 -18.7	G9 -19.0	G10 -15.2
2.40 -12.9	G10 -18.9	G10 -14.0		2.41 -2.4	2.40 -11.9	2.40 -9.0	G10 -11.7	2.40 -7.5		
2.41 -13.0	2.40 -15.8	2.40 -5.3		TR2 -12.2	2.41 -12.1	2.41 -8.9	2.41 -8.9	2.41 -8.6		
TR2 -13.0	2.41 -15.4	2.41 -5.6			TR2 -17.2	TR2 -16.7	2.41 -14.5	TR2 -13.3		
		TR2 -8.7	TR2 -10.5				TR2 -17.2	TR2 -17.2		
Linear Penalty	X	HB -2.4						HB -8.9		
		G6 -11.7					G6 -12.8	G6 -17.2	G7 -5.7	
		G7 -4.8		G6 -6.9	G6 -17.2	G6 -4.8	G9 -14.8	G7 -2.5	G7 -17.4	G9 -5.3
		G9 -4.0		G9 -3.3			TR2 -17.2	G9 -18.3	G9 -19.0	G10 -3.5
2.40 -3.4						TR2 -12.6	TR2 -17.1			
2.41 -3.8										
AL single	X						G6 -7.2		HB -5.1	
						G6 -15.9	G9 -11.0	G9 -17.7	G6 -12.3	
							TR2 -13.8	TR2 -7.7	G7 -16.5	
									G9 -19.0	
							TR2 -12.6	TR2 -12.6		
AL single old settings	X	HB -4.7	HB -10.6				HB -5.9		HB -16.1	
		G7 -2.9	G7 -5.5				G6 -12.6		G6 -15.9	
		2.40 -5.1	G10 -17.4	X		G6 -16.9	G7 -2.8	G7 -4.8	G7 -15.0	
		2.41 -9.0	2.40 -10.7				G9 -8.4	G9 -17.6	G9 -18.7	G7 -5.8
	2.41 -13.7				2.40 -3.5	TR2 -11.3	2.40 -9.5	2.41 -12.7		
					2.41 -6.5		2.41 -12.7	TR2 -16.7		
					TR2 -17.0		TR2 -16.7			
AL many old settings	G7 -17.1	HB -10.6	HB -15.2	HB -4.5			HB -13.0	HB -9.0	HB -17.0	HB -8.2
		G7 -17.2	G7 -17.2	G7 -17.2			G7 -17.2	G7 -17.4	G7 -17.4	G7 -17.2
		2.40 -10.0	G9 -6.0	G9 -5.4	X		G9 -15.2	G9 -18.5	G9 -19.0	G9 -6.8
		2.41 -13.4	G10 -9.0	G9 -5.4			2.40 -9.2	2.40 -4.7	2.40 -15.4	2.40 -3.0
	2.40 -14.9	2.41 -2.5			2.41 -12.1	2.41 -7.6	2.41 -15.3	2.41 -6.5		
	2.41 -16.0				TR2 -17.2	TR2 -12.4	TR2 -17.1			
AL many	G7 -16.9 G10 -11.4	HB -13.9	HB -16.5	HB -9.2			HB -15.5	HB -12.6	HB -17.2	HB -12.3
		G7 -17.2	G6 -4.5	G7 -17.1			G6 -14.1	G6 -4.5	G6 -16.5	G7 -12.3
		G10 -12.5	G7 -17.2	G9 -7.6		G6 -17.0	G7 -17.2	G7 -17.4	G7 -17.4	G7 -17.2
		2.40 -14.1	G9 -8.8	G10 -13.0		2.40 -3.8	G9 -16.3	G9 -18.6	G9 -19.0	G9 -10.4
2.41 -15.3	G10 -18.1	2.40 -5.3		2.41 -2.7	G10 -11.2	G10 -11.0	G10 -12.8	G10 -13.4		
	2.40 -16.4	2.41 -6.7			2.40 -13.2	2.40 -10.8	2.40 -16.7	2.40 -8.0		
	2.41 -16.6				2.41 -14.6	2.41 -11.2	2.41 -16.4	2.41 -10.5		
					TR2 -17.0	TR2 -8.9	TR2 -16.4			
MM-AL many	G10 -12.1		HB -3.3						HB -12.3	
			G7 -5.7						G6 -3.8	
			G10 -18.9	G10 -17.1	G6 -13.5				G7 -17.4	G7 -6.5
			2.40 -7.4				X	G9 -16.0	G9 -17.4	G10 -17.1
	2.41 -7.4						G10 -16.8			
							2.40 -5.1			
							2.41 -4.6			
MM-AL many Restricted CP, f -evals	G10 -8.4	G10 -13.0	HB -5.5				G6 -5.7		HB -11.9	
		2.40 -3.1	G10 -17.5	G10 -13.5	G6 -8.0		2.41 -3.2	X	G6 -7.8	
		2.41 -6.5	2.40 -10.3				TR2 -9.0		G10 -13.3	G10 -13.5
			2.41 -12.3						2.40 -8.5	
							2.41 -11.2			
							TR2 -7.0			
MM-AL many Restricted CP, g -evals	X								X	
AL many Restricted CP	X	HB -2.7	HB -8.7				HB -3.6		HB -15.6	
		G6 -4.6	G6 -14.3				G6 -17.2	G6 -15.3	G6 -17.2	
		2.40 -4.1	G10 -17.3	G6 -12.1	G6 -17.2	G6 -10.8	G9 -11.6	G9 -17.8	G9 -19.0	X
		2.41 -4.2	2.40 -11.0				2.40 -2.6	TR2 -7.8	2.40 -9.4	
	2.41 -10.0				TR2 -16.8		2.41 -7.7			
							TR2 -16.0			

Table 2: Comparing the different algorithms’ on the proposed benchmark test set. If the test is significant, meaning algorithm in row is better than algorithm in column ($p < 0.05/\beta$) for a test problem, we display the problem name and $\log_{10} p$. If $p < 10^{-3}/\beta$, it is displayed as bold text.