Analysis of Randomized Adaptive Algorithms for Black-Box Continuous Constrained Optimization

Asma Atamna

NNT : 2017SACLS010

Thèse de doctorat de l'Université Paris-Saclay préparée à l'Université Paris-Sud

École doctorale n°580
Sciences et Technologies de l'Information et de la Communication
Spécialité de doctorat : Informatique

par

Mme Asma Atamna

Analysis of Adaptive Randomized Algorithms for Black-Box Continuous Constrained Optimization

Thèse présentée et soutenue à Orsay, le 25 janvier 2017.

Composition du Jury :

M.	Serge Gratton	Professeur INP Toulouse, ENSEEIHT	(Président du jury)
M.	Christian Igel	Professeur University of Copenhagen	(Rapporteur)
M.	Rodolphe Le Riche	Directeur de recherche CNRS, École des Mines de Saint-Etienne	(Rapporteur)
M.	Dirk V. Arnold	Professeur Dalhousie University	(Examinateur)
M.	J. Frédéric Bonnans	Directeur de recherche Inria Saclay, CMAP, École Polytechnique	(Examinateur)
M.	Tobias Glasmachers	Junior professor Ruhr-Universität Bochum	(Examinateur)
Mme	Anne Auger	Chargée de recherche Inria Saclay	(Co-encadrante de thèse)
M.	Nikolaus Hansen	Directeur de recherche Inria Saclay	(Directeur de thèse)

Abstract

We investigate various aspects of adaptive randomized (or stochastic) algorithms for both constrained and unconstrained black-box continuous optimization.

The first part of this thesis focuses on step-size adaptation in unconstrained optimization. We first present a methodology for assessing efficiently a step-size adaptation mechanism that consists in testing a given algorithm on a minimum set of functions, each reflecting a particular difficulty that an efficient step-size adaptation algorithm should overcome. We then benchmark two step-size adaptation mechanisms on the well-known BBOB (black-box optimization benchmarking) noiseless testbed and compare their performance to the one of the state-of-the-art evolution strategy (ES), CMA-ES, with cumulative step-size adaptation.

In the second part of this thesis, we investigate linear convergence of a $(1+1)$-ES and a general step-size adaptive randomized algorithm on a linearly constrained optimization problem, where an adaptive augmented Lagrangian approach is used to handle the constraints. To that end, we extend the Markov chain approach used to analyze randomized algorithms for unconstrained optimization to the constrained case. We prove that when the augmented Lagrangian associated to the problem, centered at the optimum and the corresponding Lagrange multipliers, is positive homogeneous of degree 2, then for algorithms enjoying some invariance properties, there exists an underlying homogeneous Markov chain whose stability (typically positivity and Harris-recurrence) leads to linear convergence to both the optimum and the corresponding Lagrange multipliers. We deduce linear convergence under the aforementioned stability assumptions by applying a law of large numbers for Markov chains. We also present a general framework to design an augmented-Lagrangian-based adaptive randomized algorithm for constrained optimization, from an adaptive randomized algorithm for unconstrained optimization.

Résumé

On étudie dans cette thèse différents aspects des algorithmes stochastiques adaptatifs pour l'optimisation numérique boîte-noire, dans les cas avec et sans contraintes. On s'intéresse en particulier à la famille des stratégies d'évolution (ES) dont l'algorithme CMA-ES (covariance matrix adaptation evolution strategy) est reconnu comme la référence en optimisation stochastique numérique. Plus précisément, on aborde deux problèmes ouverts liés aux stratégies d'évolution et formulés ci-dessous :

(i) Existe-t-il une stratégie optimale pour adapter le step-size? Et comment évaluer et comparer efficacement des stratégies d'adaptation du step-size?

(ii) Comment gérer les constraintes efficacement? En particulier, la convergence linéaire obtenue dans le cas sans contraintes peut-elle être préservée sur des problèmes simples avec contraintes?

Nos travaux pour tenter de répondre à ces questions sont présentés dans le présent manuscrit, qui s'organise comme suit : le chapitre 1 motive les thèmes de recherche abordés et résume les contributions. Le chapitre 2 introduit quelques notions importantes de la théorie des chaines de Markov à temps discret et états continus, telles que la positivité, l'irréductibilité, la récurrence et la loi des grands nombres généralisée aux chaines de Markov. Ces notions sont ensuite utilisées pour analyser la convergence d'algorithmes stochastiques adaptatifs pour l'optimisation avec contraintes.

Le chapitre 3 est consacré à l'optimisation numérique boîte-noire. Dans un premier temps, les caractéristiques qui rendent un problème difficile à optimiser sont discutées. Ensuite, les principaux algorithmes d'optimisation numérique boîte-noire sont rappelés; l'accent est mis sur les algorithmes évolutionnaires et, en particulier, sur les stratégies d'évolution, qui sont modélisées par une séquence d'états et une fonction de transition qui donne le nouvel état à partir de l'état courant de l'algorithme. Ce chapitre introduit notamment l'importante notion de "convergence linéaire", et un exemple illustrant l'analyse de la convergence linéaire d'un ES en utilisant la théorie des chaines de Markov y est donné, pour le cas sans contraintes.

Le chapitre 4 présente une vue d'ensemble des méthodes de gestion des contraintes en optimisation mathématique et dans les algorithmes évolutionnaires. Quelques définitions

classiques telles que la faisabilité, la notion de cône critique et les conditions de faisabilité (notamment les conditions dites de Karush-Kuhn-Tucker), y sont aussi rappelées.

Le chapitre 5 présente la première partie de nos contributions, qui traite de l'adaptation du step-size dans les ES pour l'optimisation sans contraintes (problème ouvert (i)). On commence par présenter une méthodologie pour évaluer efficacement un mécanisme d'adaptation du step-size qui consiste à tester un algorithme donné sur un ensemble minimal de fonctions, dont chacune reflète une difficulté particulière qu'un mécanisme efficace d'adaptation du step-size doit être en mesure de résoudre. On compare ensuite les performances de trois méthodes d'adaptation du step-size—dont la méthode état-de-l'art CMA-ES avec "cumulative step-size adaptation"—sur le testbed non bruité BBOB (black-box optimization benchmarking).

Le chapitre 6 rassemble nos contributions dans le domaine de l'optimisation boîte-noire avec contraintes (problème ouvert (ii)). On analyse la convergence linéaire d'un (1+1)-ES et d'un algorithme général d'adaptation du step-size dans le cadre d'un problème d'optimisation avec contraintes linéaires, gérées par une approche Lagrangien augmenté adaptative. Pour ce faire, on étend l'analyse par chaînes de Markov conduite dans le cas de l'optimisation sans contraintes au cas avec contraintes. On montre que si le Lagrangien augmenté correspondant au problème et centré en l'optimum et en les multiplicateurs de Lagrange associés, est positivement homogène de degré 2, alors—pour des algorithmes présentant des propriétés d'invariance—il existe une chaîne de Markov homogène dont la "stabilité" implique la convergence linéaire de l'algorithme vers l'optimum et les multiplicateurs de Lagrange associés (par "stabilité", on entend positivité et Harris-récurrence). La convergence linéaire est déduite en appliquant une loi des grands nombres pour les chaînes de Markov, sous l'hypothèse de la stabilité. On présente ensuite une approche générale pour construire un algorithme stochastique adaptatif avec une approche Lagrangien augmenté à partir d'un algorithme stochastique adaptatif pour l'optimisation sans contraintes.

Enfin, une synthèse des résultats obtenus et quelques perspectives sont présentées dans le chapitre 7.

Acknowledgments

First and foremost, I would like to thank my advisors, Anne Auger and Nikolaus Hansen, for giving me the opportunity to pursue a Ph.D. thesis and for introducing me to research. This work would not have been possible without their guidance, support, kindness, and dedication. It has been an honor to be their Ph.D. student.

I deeply thank Christian Igel and Rodolphe Le Riche for kindly accepting to review my manuscript and for their insightful comments and suggestions. I would also like to thank the other members of my committee, Dirk V. Arnold, J. Frédéric Bonnans, Tobias Glasmachers, and Serge Gratton, for their interest in my work and their invaluable feedback.

I am grateful to Marc Schoenauer and Michèle Sebag for their warm welcome into the TAO team, where I had the chance to meet amazing people. Special thanks to Alexandre, Antoine, Aurélien, Daniela, Edgar, Gaétan, Guillaume, Jérémy, Luigi, Manh, Marie-Liesse, Nacim, Nicolas, Olga, Ouassim, Phillipe, Pierre-Yves, Riad, and Sandra for making this three-year journey even more enjoyable.

I would like to thank Youhei Akimoto, Dimo Brockhoff, Dejan Tušar, and Tea Tušar with whom I had the pleasure to collaborate on the NumBBO project and exchange valuable ideas during our group meetings.

To my dearest friends Fatima and Marco, thank you for being such a huge source of strength and positivity for me.

Lastly, I would like to thank my family for their unconditional love and unfailing support. Mom, Dad, Lina, Mehdi, I love you more than anything in this world.

Table of contents

Notations and Acronyms

$[\mathbf{M}]_{ij}$	element in the ith row and the jth column of a matrix $\mathbf{M}$
$[\mathbf{x}]_i$	ith coordinate of a vector $\mathbf{x}$
$\mathbf{0}$	vector of all-zeros $\mathbf{0} = (0, \cdots, 0)^\intercal \in \mathbb{R}^n$
$\mathbf{C}^{1/2}$	square root of a matrix $\mathbf{C}$, satisfies $\mathbf{C}^{1/2}(\mathbf{C}^{1/2})^\intercal = \mathbf{C}$
$\circ$	function composition operator
CR	convergence rate
n	dimension of the search space
$\emptyset$	empty set
$\mathbf{I}_{n\times n}$	identity matrix in $\mathbb{R}^{n\times n}$
$\mathbf{1}_{\{A\}}$	indicator function that returns 1 if A is true and 0 otherwise
$\ln(.)$	natural logarithm function
$\|\mathbf{x}\|$	Euclidean norm of a vector $\mathbf{x}$
$\mathcal{N}(\boldsymbol{m}, \mathbf{C})$	multivariate normal distribution of mean $\boldsymbol{m}$ and covariance matrix $\mathbf{C}$
$\mathbb{N}$	set of natural numbers
$\mathbb{N}_>$	set of natural numbers without zero
$\nabla^2_{\mathbf{xx}}$	second-order derivative with respect to $\mathbf{x}$
$\nabla_{\mathbf{x}}$	derivative with respect to $\mathbf{x}$
m	number of constraints

$\Pr(X)$ "probability of event X"

$\mathbb{R}$ set of real numbers

$\mathbb{R}^+$ set of positive real numbers

$\mathbb{R}^+_>$ set of strictly positive real numbers

$\sim$ equality in distribution

$\mathbf{x}^\mathsf{T}$ transpose of $\mathbf{x}$

$A := B$ "A is defined as B"

$A \setminus B$ complement of B with respect to A, represents the elements of a set A that are not in B

$E_\pi(X)$ expectation of random variable $X \sim \pi$

$|x|$ absolute value of x

a.s. almost surely

aRT average runtime

CB comparison-based

CMA covariance matrix adaptation

CMA-ES covariance matrix adaptation evolution strategy

CSA cumulative step-size adaptation

DFO Derivative-free optimization

EA evolutionary algorithm

ERT expected running time

ES evolution strategy

FVF function-value-free

i.i.d. independent identically distributed

KKT Karush-Kuhn-Tucker

LICQ linear independence constraint qualification

LLN law of large numbers

MFCQ Mangasarian-Fromovitz constraint qualification

MSR median success rule step-size adaptation

TPA two-point step-size adaptation

w.l.o.g. without loss of generality

Chapter 1

Introduction

Black-box continuous optimization problems are often encountered in practice; they consist in minimizing a function f defined on a search space $\mathscr{X} \subseteq \mathbb{R}^n$, without knowing any information on f but the value $f(\mathbf{x})$ for a given point $\mathbf{x} \in \mathscr{X}$. Randomized (or stochastic) algorithms have been applied successfully to a wide range of real-world continuous problems and a particular family of randomized algorithms, the evolution strategies (ESs) [48], have proven to be particularly efficient as demonstrated by the performance of the state-of-the-art ES, the covariance matrix adaptation evolution strategy (CMA-ES) [53], on practical problems, including ill-conditioned and non-separable ones. Given the current estimate $\mathbf{X}_t$ of the optimum, an ES samples new candidate solutions $\mathbf{X}_{t+1}^i$, $i = 1, \cdots, \lambda$, according to

$$\mathbf{X}_{t+1}^i = \mathbf{X}_t + \sigma_t \mathscr{N}_t^i(\mathbf{0}, \mathbf{C}_t) \ ,$$

where $\mathscr{N}_t^i(\mathbf{0}, \mathbf{C}_t)$ denotes a multivariate normal distribution of mean $\mathbf{0}$ and covariance matrix $\mathbf{C}_t$ and $\sigma_t > 0$ is the *step-size* and determines the "width" of the distribution of $\mathbf{X}_{t+1}^i$. The step-size and the covariance matrix are typically adapted so as to increase the likelihood of "good" solutions; consequently, ESs are observed to converge *linearly* on a large set of unconstrained problems, that is, the distance of the current solution $\mathbf{X}_t$ to the optimum of f, $\mathbf{x}_{\text{opt}}$, decreases linearly in log-scale. The performance of an ES therefore directly relies on how the step-size and the covariance matrix are adapted. The adaptation of the covariance matrix in moderate dimensions seems to be a solved problem and can be achieved with CMA-ES for example. On the other hand, the question of how to adapt the step-size efficiently is still open, and we approach this problem in Chapter 5.

Real-world optimization problems are often *constrained*, and despite the ongoing research efforts in the field of constrained optimization, the question of how to handle constraints properly is still open. In this thesis, we aim at designing a practical randomized algorithm for

constrained optimization with a sound theoretical background. Following [5], we argue that an effective constraint handling algorithm should converge linearly on "simple" constrained problems, such as convex quadratic problems subject to a single linear constraint, at a reasonable computational cost.

Linear convergence is an important aspect in unconstrained optimization and many randomized algorithms for black-box optimization are designed with the purpose of converging linearly on the widest possible range of problems. In the unconstrained case, linear convergence of adaptive randomized algorithms—and ESs in particular— is commonly analyzed with tools from the Markov chain theory [67]; the general approach consists in constructing a homogeneous Markov chain whose stability (usually positivity and Harris-recurrence) will lead to linear convergence through the application of a law of large numbers for Markov chains. Such a Markov chain typically exists for translation-invariant and scale-invariant ESs on the class of scaling-invariant functions [10, 14]. As for constrained optimization, a $(1+1)$-ES with an augmented Lagrangian constraint handling approach is presented in [5] for the case of one inequality constraint; the algorithm is observed to converge linearly on two convex quadratic functions when the constraint is linear, without the need to adapt the covariance matrix. A part of our work consists in analyzing linear convergence of the algorithm in [5] using the Markov chain approach described above. In the light of the obtained results, we present a practical general step-size adaptive randomized algorithm with augmented Lagrangian constraint handling, for $m \geq 1$ inequality constraints, and apply the same Markov chain approach to analyze its linear convergence in the case of linear constraints.

The rest of this thesis is organized as follows: we introduce general concepts of the Markov chain theory in Chapter 2 in order to give a broad idea on how stability is proven in practice. Then we present an overview of black-box continuous optimization in Chapter 3; we highlight in particular difficulties that may arise in practice and present the class of *comparison-based* adaptive randomized algorithms, which includes ESs. In Chapter 4, we present theoretical aspects of constrained optimization and review some well-known constraint handling approaches, with particular emphasis on augmented Lagrangian approaches. In Chapters 5 and 6, we present our contributions which can be divided in the two following parts:

1) Assessment of step-size adaptation mechanisms in unconstrained optimization (Chapter 5). We present two papers related to step-size adaptation in unconstrained optimization. The first paper [51] describes a methodology for assessing step-size adaptation mechanisms, which are often evaluated on too restrictive scenarios. We propose a minimal set of test

functions for a more realistic and thorough assessment and illustrate our methodology on three algorithms. In the second paper [6], we benchmark two relatively recent step-size adaptation mechanisms on the BBOB testbed [52] and discuss the results.

2) Markov chain analysis of linear convergence in constrained optimization (Chapter 6). This part details the main contributions of this thesis. In Section 6.1, we present a Markov chain analysis of the augmented-Lagrangian-based $(1+1)$-ES presented in [5] for the case of a single linear constraint. We show that if the augmented Lagrangian associated to the constrained problem, centered at the optimum and the corresponding Lagrange multiplier, is positive homogeneous of degree 2, then there is an underlying homogeneous Markov chain such that, if some stability conditions hold, the algorithm converges linearly to both the optimum and the corresponding Lagrange multiplier. To construct this Markov chain, we exploit the comparison-based aspect of the algorithm along with its translation-invariance and scale-invariance. We validate the stability empirically on the sphere function and on a moderately ill-conditioned ellipsoid function. This work was originally presented in [7]. In Section 6.2, we present a general framework for building an adaptive randomized algorithm with augmented Lagrangian constraint handling from an adaptive randomized algorithm for unconstrained optimization [8]. We define this framework for the case of one inequality constraint; however, the generalization to m constraints is straightforward. This framework is then used to construct a $(\mu/\mu_W,\lambda)$-CMA-ES with adaptive augmented Lagrangian constraint handling. The algorithm is tested on a set of problems, including ill-conditioned problems, and linear convergence is observed. In Section 6.3, we present a general step-size adaptive randomized algorithm with an adaptive augmented Lagrangian approach to handle $m \geq 1$ inequality constraints. To adapt the penalty factors of the augmented Lagrangian, we propose a generalized version of the update rule introduced in [5]. We then analyze this algorithm using a Markov chain approach: similarly to the single constraint case, we show the existence of a homogeneous Markov chain whose stability implies linear convergence on the class of functions such that the augmented Lagrangian, centered at the optimum and the corresponding Lagrange multipliers, is positive homogeneous of degree 2. Once again, the stability of the constructed Markov chain is validated numerically. This work was submitted to the workshop on Foundations of Genetic Algorithms for possible publication.

In Chapter 7, we give a general discussion and perspectives for future work.

Chapter 2

Introduction to Markov Chain Theory

Markov chain theory [67, 45] has played a central role in the analysis of linear convergence of comparison-based randomized algorithms for unconstrained optimization [21, 10, 14, 15]. By adopting a Markov chain approach, the study of linear convergence is replaced by the study of the *stability* of an underlying Markov chain. In our case, we generalize the Markov chain approach used in the unconstrained case to the constrained case as follows: we exhibit a class of functions on which the investigated algorithm can be modeled as a homogeneous Markov chain that "has a chance to be stable". If the stability of the constructed Markov chain holds, then linear convergence follows by virtue of the law of large numbers for Markov chains.

In this chapter, we introduce some important definitions and theorems of the Markov chain theory. Although most of these concepts are not explicitly used in our work, they are essential in understanding and proving the stability of a Markov chain.

2.1 Discrete-Time Markov Chains

In our context, the term *Markov chain* refers to a discrete-time sequence $(X_t)_{t\in\mathbb{N}}$ of random variables taking values in an open set $S \subset \mathbb{R}^n$, equipped with a Borel σ-algebra $\mathscr{B}(S)$, that we call the *state space*. The sequence $(X_t)_{t\in\mathbb{N}}$ satisfies the *Markov property*, that is, the conditional distribution of X_{t+1} given the past states $X_0, \cdots, X_t$, $t \in \mathbb{N}$, depends only on X_t. We consider *time homogeneous* Markov chains, where the conditional distribution of X_{t+1} given X_t is independent of t. The probabilities specifying the conditional distribution of X_{t+1} given X_t, or *transition probabilities*, are given by a *transition probability kernel* P defined as

$$P(\mathbf{x}, B) = \Pr(X_{t+1} \in B \mid X_t = \mathbf{x}) \ ,$$

with $\mathbf{x} \in S$ and $B \in \mathscr{B}(S)$ a measurable set. The probability transition kernel P satisfies the following two conditions:

(i) The function $P(\mathbf{x}, .)$ for all $\mathbf{x} \in S$ is a probability measure.

(ii) The function $P(., B)$ for all $B \in \mathscr{B}(S)$ is a measurable function.

Given $X_t = \mathbf{x}$, the probability P^k of hitting a subset $B \subseteq S$ in k steps ($k \geq 2$), starting from $\mathbf{x}$, is defined inductively as

$$P^k(\mathbf{x}, B) = \Pr(X_{t+k} \in B \mid X_t = \mathbf{x}) = \int_S P(\mathbf{x}, d\mathbf{y}) P^{k-1}(\mathbf{y}, B) \ ,$$

where $P^1 := P$.

2.2 Invariant Probability Measure

Let π be a probability measure on the state space S and let assume that $X_t \sim \pi$. Then, the distribution πP of X_{t+1} is given by

$$\pi P(B) = \int_S \pi(d\mathbf{x}) P(\mathbf{x}, B) \ .$$

We say that π is *invariant* if

$$\pi(B) = \int_S \pi(d\mathbf{x}) P(\mathbf{x}, B) \ ,$$

that is, $\pi P = \pi$. Informally, this means that if $X_0 \sim \pi$, then $X_t \sim \pi$ for all $t \in \mathbb{N}$. If an invariant probability measure exists for a Markov chain, we say that this Markov chain is *positive*.

2.3 Irreducibility

We say that a Markov chain is φ-*irreducible* if there exists a nonzero measure φ on the state space such that for any $\mathbf{x} \in S$ and for any measurable set $B \in \mathscr{B}(S)$ such that $\varphi(B) > 0$, $\sum_{k \in \mathbb{N}_>} P^k(\mathbf{x}, B) > 0$. Hence, φ-irreducibility ensures that all φ-positive sets are reachable from anywhere in the search space. Note that φ is an arbitrary measure that is not necessarily an invariant probability measure. Indeed, a Markov chain that does not admit an invariant probability measure can still be φ-irreducible.

If a Markov chain is φ-irreducible, then there exists a *maximal irreducibility measure* ψ [67, Proposition 4.2.2] that dominates any other irreducibility measure[1].

2.4 Harris-Recurrence

We adopt the definition of Harris-recurrence in [67]. Consider a ψ-irreducible Markov chain. We say that a measurable set $B \in \mathscr{B}(S)$ is *Harris-recurrent* if for all $\mathbf{x} \in B$

$$\Pr\left(\sum_{t\in\mathbb{N}_>} \mathbf{1}_{\{X_t\in B\}} = \infty \mid X_0 = \mathbf{x}\right) = 1 \quad ,$$

that is, starting from some point $\mathbf{x} \in B$, the Markov chain will return an infinite number of times to B almost surely. By extension, a ψ-irreducible Markov chain is *Harris-recurrent* if all ψ-positive sets are Harris-recurrent.

2.5 Law of Large Numbers for Markov Chains

The following theorem generalizes the law of large numbers (LLN) for independent identically distributed (i.i.d.) random variables to Markov chains. It states that the LLN holds for Markov chains if some stability properties, namely positivity and Harris-recurrence, are satisfied by the Markov chain.

Theorem 1 (Theorem 17.0.1 from [67])**.** *Let X be a positive Harris-recurrent chain with invariant probability π. Then the LLN holds for any function q such that $\pi(|q|) = \int |q(\boldsymbol{x})|\pi(d\boldsymbol{x}) < \infty$, that is, for any initial state X_0, $\lim_{t\to\infty} \frac{1}{t}\sum_{k=0}^{t-1} q(X_k) = \pi(q)$ almost surely.*

If Harris-recurrence holds, the mean of the sample $q(X_0), \cdots, q(X_t)$, when $t \to \infty$, converges almost surely to the expected value with respect to the invariant probability measure π, given the function q is integrable.

2.6 Periodicity

If a Markov chain is ψ-irreducible, its search space can be partitioned into sets $D_0, \cdots, D_{d-1}$ and N such that [45]

(i) $P(\mathbf{x}, D_i) = 1$, for $\mathbf{x} \in D_j$ and $j = i - 1 \mod d$.

[1] A measure ψ dominates another measure φ if for any measurable set A, $\psi(A) = 0$ implies $\varphi(A) = 0$.

(ii) $\psi(N) = 0$.

We say that the Markov chain is *aperiodic* if $d = 1$ and *periodic* if $d > 1$.

2.7 Ergodicity

A positive Harris-recurrent Markov chain with invariant probability measure π is said to be *ergodic* if for any initial condition $X_0 = \mathbf{x} \in S$,

$$\|P^t(\mathbf{x},.) - \pi\| \xrightarrow[t\to\infty]{} 0 \quad .$$

If $\|P^t(\mathbf{x},.) - \pi\|$ converges to 0 at a geometric rate, we say that the Markov chain in *geometrically ergodic*. In [67], a slightly different definition is given: a positive Harris-recurrent Markov chain with invariant probability measure π is geometrically ergodic if there exists a constant $r > 1$ such that for any initial condition $X_0 = \mathbf{x} \in S$,

$$\sum_{t\in\mathbb{N}_>} r^t \|P^t(\mathbf{x},.) - \pi\| < \infty \quad .$$

2.8 Small and Petite Sets

A set $C \in \mathscr{B}(S)$ is a *small set* if there exist $m \in \mathbb{N}_>$ and a nonzero measure ν_m such that for all $\mathbf{x} \in C$ and for all measurable sets $B \in \mathscr{B}(S)$,

$$P^m(\mathbf{x},B) \geq \nu_m(B) \quad .$$

A set $C \in \mathscr{B}(S)$ is a *petite set* if there exist a probability distribution α on $\mathbb{N}$ and a nonzero measure ν_α such that for all $\mathbf{x} \in C$ and for all measurable sets $B \in \mathscr{B}(S)$,

$$\sum_{t\in\mathbb{N}} \alpha(t) P^t(\mathbf{x},B) \geq \nu_\alpha(B) \quad ,$$

where $P^0(\mathbf{x},B)$ represents a Dirac distribution on $\{\mathbf{x}\}$. In practice, compact sets are often small sets [25].

2.9 Feller Chains and T-Chains

A φ-irreducible Markov chain is called a *Feller chain* if the function $P(.,O)$ is lower semi-continuous, for all open sets O in the search space. If a Markov chain is a *T-chain*, there exists a distribution α on $\mathbb{N}$ and a kernel $T(\mathbf{x},B)$ such that

(i) for all $\mathbf{x} \in S$ and for all $B \in \mathscr{B}(S)$, $\sum_{t\in\mathbb{N}} \alpha(t)P^t(\mathbf{x},B) \geq T(\mathbf{x},B)$,

(ii) for all $\mathbf{x} \in S$, $T(\mathbf{x},S) > 0$.

These two notions are used to identify small and petite sets of a Markov chain.

2.10 Drift Conditions

As stated in Theorem 1, the LLN generalizes to Markov chains when positivity and Harris-recurrence hold. Since positivity and Harris-recurrence imply irreducibility, the first step is to verify whether the considered Markov chain is irreducible. Drift conditions come into play to show recurrence, positivity, and ergodicity. They are expressed as a function of the drift (or potential) function $V : S \to \mathbb{R}^+$ and the expectation

$$PV(\mathbf{x}) := E(V(X_{t+1}) \mid X_t = \mathbf{x}) \ .$$

If the sets $\{\mathbf{x} \in S \mid V(\mathbf{x}) \leq r\}$ are petite for any real number r, we say that V is *unbounded off petite sets*. The following drift condition ensures Harris-recurrence.

Theorem 2 (Theorem 9.1.8 from [67])**.** *Let $(X_t)_{t\in\mathbb{N}}$ be a ψ-irreducible Markov chain. If there exists a petite set $C \in \mathscr{B}(S)$ and a non-negative function V that is unbounded off petite sets such that*

$$PV(\boldsymbol{x}) \leq V(\boldsymbol{x}), \textit{ for all } \boldsymbol{x} \notin C \ ,$$

then the Markov chain is Harris-recurrent.

A stronger drift condition, the so-called *geometric drift condition* [67, Theorem 15.0.1], states that given a φ-irreducible aperiodic Markov chain, if there exists a function $V \geq 1$, a petite set $C \in \mathscr{B}(S)$, constants $\beta > 0$ and $b < \infty$ such that

$$PV(\mathbf{x}) - V(\mathbf{x}) \leq -\beta V(\mathbf{x}) + b\mathbf{1}_{\{\mathbf{x}\in C\}}, \ \mathbf{x} \in S \ ,$$

then—among other consequences—the Markov chain is positive recurrent with invariant probability measure π.

The drift function is often chosen as the objective function f plus some constant in proofs.

Chapter 3

Black-Box Continuous Optimization: an Overview

Continuous (or numerical) optimization considers the minimization[1] of functions of the form $f : \mathscr{X} \subseteq \mathbb{R}^n \to \mathbb{R}$ defined on a subset $\mathscr{X}$ of $\mathbb{R}^n$ called the *search space*, where $n \in \mathbb{N}_>$ is the dimension of the search space and where f is referred to as the *objective function*. A solution to this optimization problem is called the *optimum*—or the *minimum*—of f[2].

In black-box optimization, the objective function f is minimized in a *black-box scenario*, that is, the only information on f available during the optimization process is $f(\mathbf{x})$ for a given point $\mathbf{x} \in \mathscr{X}$ (zero-order information). In particular, no higher-order information is available, such as the gradient (first-order information) or the Hessian (second-order information) of f. This is a common scenario in real-world applications where the f-value (also fitness or quality) of a solution is often obtained via an executable file whose source code is not available or by running a simulation model.

In this chapter, we discuss various aspects of black-box continuous optimization: we highlight some difficulties that might be encountered when optimization a continuous function in Section 3.1. Then, we present some of the best-known algorithms for black-box continuous optimization in Sections 3.2 and 3.3. These algorithms are known generically as *derivative-free optimization* (DFO) algorithms, as opposed to algorithms exploiting derivatives such as Newton and quasi-Newton methods [37, 39]. We distinguish *deterministic* algorithms and *stochastic* algorithms, as well as *function-value-free* (FVF)—or *comparison-based* (CB)—

[1] We consider minimization w.l.o.g. Indeed, maximizing f is equivalent to minimizing $-f$. We may also refer to minimization simply as optimization.

[2] Due to the nature of the search space, practical algorithms for numerical optimization can only approximate the optimum. In discrete optimization, however, exact solutions can be found.

algorithms, which use f-values of the generated solutions only through comparisons, and *value-based* algorithms which explicitly use f-values.

3.1 Difficulties Related to Continuous Optimization

We discuss in the following five characteristics that make an optimization problem difficult to solve [11].

Ruggedness Informally, we say that a function is *rugged* if its graph is "uneven". Ruggedness can be due to different reasons such as *multi-modality*, that is, the presence of many local optima. Another possible reason is that the function under consideration is *discontinuous*, *non-differentiable*, or *noisy* (in which case two function evaluations of the same point $\mathbf{x}$ return different values). To tackle such problems, an optimization algorithm generally needs many function evaluations to capture the structure of the function at hand.

Non-separability An objective function $f(x_1, \cdots, x_n)$ is *separable* if its optimal value for each coordinate, $[\mathbf{x}_{\text{opt}}]_i$, $i = 1, \cdots, n$, can be obtained by optimizing $f(\bar{x}_1, \cdots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \cdots, \bar{x}_n)$ for any fixed values $\bar{x}_1, \cdots, \bar{x}_{i-1}, \bar{x}_{i+1}, \cdots, \bar{x}_n$. More formally, for all $i \in \{1, \cdots, n\}$,

$$[\mathbf{x}_{\text{opt}}]_i = \arg\min_{x_i \in \mathbb{R}} f(\bar{x}_1, \cdots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \cdots, \bar{x}_n) \ ,$$

for all $\bar{x}_1, \cdots, \bar{x}_{i-1}, \bar{x}_{i+1}, \cdots, \bar{x}_n \in \mathbb{R}$. This means that separable functions can be optimized by solving n one-dimensional problems. On non-separable functions, however, an algorithms needs to take the dependencies between variables into account.

Ill-conditioning For convex quadratic functions of the form

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^\intercal \mathbf{H} (\mathbf{x} - \bar{\mathbf{x}}) \ , \tag{3.1}$$

with $\mathbf{H} \in \mathbb{R}^{n \times n}$ a symmetric positive-definite matrix and $\bar{\mathbf{x}} \in \mathbb{R}^n$, the *conditioning* is given by the condition number of the Hessian $\mathbf{H}$, and its square root corresponds to the ratio between the lengths of the largest and the smallest axis of the ellipsoidal contour lines of f. In practice, a problem is considered ill-conditioned if its conditioning is larger than 10^5. The difficulty when dealing with ill-conditioned problems resides in the different scaling of changes in f-values along different axes.

Remark 1. If $\mathbf{H}$ in (3.1) is the identity matrix $\mathbf{I}_{n\times n}$, f corresponds to the *sphere function*. If $\mathbf{H}$ is a diagonal matrix with diagonal elements $\alpha^{\frac{i-1}{n-1}}$, $i = 1,\cdots,n$, $\alpha > 0$, then f corresponds to the *ellipsoid function* and α is the condition number of $\mathbf{H}$.

Dimensionality The volume of the search space increases exponentially with the dimension: this is what we call the "curse of dimensionality". Consequently, an algorithm that performs well in small dimensions may not perform as good in large dimensions where a more effective exploration of the search space is required.

Constraints Constrained optimization problems are very common in practice. The presence of constraints adds to the difficulty of the original problem in that an algorithm has to find *feasible* solutions (i.e. solutions that satisfy the constraints) possibly while dealing with the aforementioned difficulties. Therefore, the efficiency of an algorithm directly depends on the one of its constraint handling mechanism.

3.2 Deterministic Algorithms

In this section, we present very briefly three deterministic algorithms for black-box continuous optimization: the Nelder-Mead method which is also FVF, pattern search methods, and zero-order trust-region methods. Deterministic algorithms present the advantage of being simple to understand by the user; they are also easier to analyze compared to stochastic algorithms. However, they typically lack the ability of *exploration*, which is particularly important in a black-box context.

3.2.1 Nelder-Mead Method

The Nelder-Mead method (also known as downhill method) [72] is one of the simplest and best-known deterministic DFO algorithms for unconstrained optimization. It is an iterative algorithm that evolves, at each iteration t, $n+1$ points $\mathbf{x}_t^1,\cdots,\mathbf{x}_t^{n+1}$ corresponding to the vertices of a *simplex* (a polytope of $n+1$ vertices in dimension n) as follows: first, the vertices are ranked according to their f-values and the centroid $\mathbf{x}_t^c$ of the best n points is computed. The algorithm then tries to replace the worst vertex $\mathbf{x}_t^{n+1:n+1}$ by one of three specific points on the line between $\mathbf{x}_t^c$ and $\mathbf{x}_t^{n+1:n+1}$ sampled using *reflection*, *expansion*, or *contraction* towards the centroid. If none of these points is better than $\mathbf{x}_t^{n+1:n+1}$, the simplex is shrunk towards the best point $\mathbf{x}_t^{1:n+1}$. Despite its simplicity, the Nelder-Mead method shows serious shortcomings as shown in [66], where the algorithm fails to converge to a stationary point on

a strictly convex function in dimension 2, and the best-known proof of convergence holds only for dimension 1 on strictly convex functions [62].

3.2.2 Pattern Search Methods

Pattern search methods [58, 88, 9] are deterministic iterative DFO algorithms that date back to the 1950s [35]. The first formal definition of these methods was presented in [38]. The principle of pattern search methods is the following: given a function f to minimize, solutions are sampled around the current best solution $\mathbf{x}_t$ according to a certain *pattern*, that is, a set of search directions. If the sampling is successful (a better solution is found), the current best solution is updated and the process is repeated around the new solution. Otherwise, the pattern is reduced and the process is repeated around $\mathbf{x}_t$. The performance of a pattern search algorithm relies to a great extent on the choice of the pattern [73], which may need to be adapted between iterations.

3.2.3 Trust-Region Methods

Trust-region methods [33, 73] are deterministic iterative algorithms that use a model for the objective function in a neighborhood of the current solution, called the *trust-region*. At each iteration t, the objective function f is approximated by a model (usually quadratic) within the trust-region and this model is optimized instead of f. The trust region can be a ball of radius r_t around the current solution $\mathbf{x}_t$ [34] and depending on the quality of the new solution, r_t is either increased or decreased. In derivative-free trust-region methods, only zero-order information on f is used to compute the model, as in the state-of-the-art trust-region method NEWUOA [77] and its variants for bound constrained optimization [78, 79]. However, some trust-region methods use higher-order information for constructing the model [34, 91]. An overview of the most recent works on trust-region methods is presented in [92].

3.3 Randomized Algorithms

In randomized or stochastic algorithms, solutions are computed using random variables. This makes these algorithms naturally adapted to black-box optimization where only zero-order information on the objective function is available. Another advantage of stochastic algorithms is that they favor the exploration of the search space. We present in this section three randomized algorithms for black-box continuous optimization: pure random search, particle swarm optimization, and the family of evolutionary algorithms, with a particular focus on the latter.

3.3.1 Pure Random Search

Pure random search (PRS) is the simplest stochastic algorithm. The algorithm samples a sequence $(\mathbf{X}_t)_{t\in\mathbb{N}}$ of i.i.d. random vectors from a distribution P. The objective function f is then evaluated on each point $\mathbf{X}_t$[3] of the sequence and the point with minimal f-value is the solution returned by the algorithm. The sequence of solutions generated by PRS is proven to always converge to the optimum $\mathbf{x}_{\text{opt}}$ of the objective function at hand [85]. Nonetheless, the algorithm is very inefficient and needs on average $1/\varepsilon^n$ iterations to enter a ball of radius $\varepsilon > 0$ around the optimum [93].

3.3.2 Particle Swarm Optimization

Particle swarm optimization (PSO) [60, 83, 29, 74] is a comparison-based stochastic algorithm inspired by the behavior of bird flocks. The algorithm evolves a *swarm* of *particles* (candidate solutions), $\mathbf{X}_t^i$, $i = 1, \cdots, p$, by stochastically updating, at each iteration, the *position* ($\mathbf{X}_t^i$) and the *velocity*, $\boldsymbol{V}_t^i$, of each particle $\mathbf{X}_t^i$, by taking into consideration the best position visited by the particle, $\boldsymbol{p}_t^i$, as well as the best particle visited by the swarm so far, $\boldsymbol{g}_t$. The updates are given by [83]

$$\begin{aligned}\boldsymbol{V}_{t+1}^i &= \omega \boldsymbol{V}_t^i + \mathscr{U}(0,\phi_1) \otimes (\boldsymbol{p}_t^i - \mathbf{X}_t^i) + \mathscr{U}(0,\phi_2) \otimes (\boldsymbol{g}_t - \mathbf{X}_t^i) \ , \\ \mathbf{X}_{t+1}^i &= \mathbf{X}_t^i + \boldsymbol{V}_t^i \ ,\end{aligned}$$

where the real parameter ω is called the inertia weight, $\mathscr{U}(0,\phi_i) \in \mathbb{R}^n$ denotes a vector of n random numbers uniformly distributed in $[0,\phi_i]$, and $\otimes$ denotes the component-wise multiplication. Despite the good performance observed on separable functions (including ill-conditioned ones), PSO shows important limitations on non-separable functions [16, 54].

3.3.3 Evolutionary Algorithms

Evolutionary algorithms (EAs) [90, 41] form an important family of stochastic (or randomized) DFO algorithms and are at the heart of this work. As suggested by their name, they emulate the mechanisms of biological evolution to seek the optimum of a function $f : \mathscr{X} \subseteq \mathbb{R}^n \to \mathbb{R}$. The underlying idea behind all evolutionary algorithms consists in evolving a set of candidate solutions, called the *population*, using bio-inspired operators such as *mutation*, *selection*, and *recombination*. More precisely, at each iteration t, a population $\mathscr{P}$ of μ points of $\mathscr{X}$, called *parents*, is used to create a new population $\mathscr{O}$ of λ points, called

[3] The use of capital $\mathbf{X}$ here indicates the randomness of $\mathbf{X}_t$. Indeed, the solution $\mathbf{X}_t$ at iteration t is a random variable in the case of stochastic algorithms.

offspring, either by recombining some of the parents or by applying a random variation (mutation) to some of them. Then, the μ *fittest* individuals are selected either in $\mathscr{O}$ (*non-elitist* selection) or in $\mathscr{O} \cup \mathscr{P}$ (*elitist* selection), where the fitness of an individual is given by its f-value. At each iteration t, the EA updates the current estimate of the optimum, $\mathbf{X}_t$, possibly along with other internal variables.

We focus our attention on a particular class of EAs, the so-called *comparison-based adaptive randomized algorithms* [15, 14]. The comparison-based aspect is important because it results in desired invariance properties and allows to model such algorithms as homogeneous Markov chains on some classes of objective functions (see Section 3.4). The adaptive aspect consists in using the information provided by the population to update internal variables, or *state variables*, of the algorithm.

Following [15, 14, 11], we give a formal definition for a general comparison-based adaptive randomized algorithm. Many EAs are comparison-based and Markovian and will naturally fit in this definition. Some, on the other hand, might be more difficult to fit. Let consider a general comparison-based adaptive randomized algorithm minimizing a function f and whose state at iteration t is given by the vector $\mathbf{s}_t \in \Omega$ of all its state variables. Given a vector $\mathbf{U}_{t+1} = [\mathbf{U}_{t+1}^1, \cdots, \mathbf{U}_{t+1}^\lambda] \in (\mathbb{R}^n)^\lambda$ of random vectors $\mathbf{U}_{t+1}^i \in \mathbb{R}^n$, $i = 1, \cdots, \lambda$, such that $p_\mathbf{U}$ is the probability distribution of each $\mathbf{U}_{t+1}$, the EA can be seen as a sequence $(\mathbf{s}_t)_{t \in \mathbb{N}}$ of states defined recursively via

$$\mathbf{s}_{t+1} = \mathscr{F}^f(\mathbf{s}_t, \mathbf{U}_{t+1}) \ , \tag{3.2}$$

where $\mathscr{F} : \Omega \times (\mathbb{R}^n)^\lambda \to \Omega$ is a deterministic *transition function* and Ω is the so-called *state space*. The superscript "f" indicates the objective function under consideration. Note that the state $\mathbf{s}_t$ typically includes the current estimate of the optimum $\mathbf{X}_t$ and that both the sampling of the offspring and the selection are encoded in $\mathscr{F}$. For sampling the candidate solutions $\mathbf{X}_{t+1}^i$, we consider the solution function $Sol : \Omega \times \mathbb{R}^n \to \mathbb{R}^n$ such that

$$\mathbf{X}_{t+1}^i = Sol(\mathbf{s}_t, \mathbf{U}_{t+1}^i), \ i = 1, \cdots, \lambda \ . \tag{3.3}$$

The candidate solutions are then ordered according to their f-values using the operator *Ord* as follows

$$\varsigma = Ord(f(\mathbf{X}_{t+1}^i)_{i=1,\cdots,\lambda}) \ , \tag{3.4}$$

where ς is the permutation that contains the indices of the ranked candidate solutions. More formally, for any λ real numbers $z_1, \cdots, z_\lambda$, $\varsigma = Ord(z_1, \cdots, z_\lambda)$ satisfies

$$z_{\varsigma(1)} \leq \cdots \leq z_{\varsigma(\lambda)} \ . \tag{3.5}$$

For a comparison-based algorithm, only the ranking of the candidate solutions matters when computing the new state $\mathbf{s}_{t+1}$. In this case, the transition function $\mathscr{F}$ can be defined as

$$\mathscr{F}^f(\mathbf{s}_t, \mathbf{U}_{t+1}) = \mathscr{G}(\mathbf{s}_t, \varsigma * \mathbf{U}_{t+1}) \ , \tag{3.6}$$

where the function $\mathscr{G} : \Omega \times (\mathbb{R}^n)^\lambda \to \Omega$ and where the operator "*" applies the permutation ς to $\mathbf{U}_{t+1}$ such that

$$\varsigma * \mathbf{U}_{t+1} = [\mathbf{U}_{t+1}^{\varsigma(1)}, \cdots, \mathbf{U}_{t+1}^{\varsigma(\lambda)}] \ .$$

If the algorithm under consideration is not comparison-based, the transition function $\mathscr{F}$ can be written as

$$\mathscr{F}^f(\mathbf{s}_t, \mathbf{U}_{t+1}) = \mathscr{T}(\mathbf{s}_t, \varsigma * \mathbf{U}_{t+1}, [f(\mathbf{X}_{t+1}^1), \cdots, f(\mathbf{X}_{t+1}^\lambda)]) \ , \tag{3.7}$$

with $\mathscr{T} : \Omega \times (\mathbb{R}^n)^\lambda \times \mathbb{R}^\lambda \to \Omega$.

In the following, we present three of the best-known EAs: genetic algorithms, differential evolution, and evolution strategies. We focus our attention on evolution strategies since all the algorithms tested in this work are evolution strategies. In particular, we connect evolution strategies to the general definition of a comparison-based adaptive randomized algorithm presented above through some examples.

Genetic Algorithms

Genetic algorithms (GAs) [71, 46] constitute an important family of EAs for discrete optimization. They were first introduced in [57] for problems defined on a binary search space of the form $\mathscr{X} = \{0,1\}^n$. GAs iteratively optimize the objective function by evolving a population of candidate solutions and use selection, mutation, and another operator called *crossover* to create new candidate solutions. In the context of discrete optimization, mutation consists in randomly flipping some bits of a candidate solution while crossover consists in exchanging some of the bits of two candidate solutions, thereby mimicking biological recombination. Although originally designed for discrete optimization, adaptations of GAs to continuous optimization problems can be found in the literature [70, 55, 89].

Differential Evolution

Differential evolution (DE) is a comparison-based EA first introduced in [86]. A population of candidate solutions is evolved and at each iteration, new individuals are added to the population by using a particular mutation operator. In its simplest form, one iteration of DE can be described as follows:

(i) Sample p candidate solutions.

(ii) For each candidate solution $\mathbf{X}_t^i$, choose randomly three different points $\mathbf{A}_t$, $\mathbf{B}_t$, and $\mathbf{C}_t$ in the population and compute $\mathbf{Z}_{t+1} = \mathbf{A}_t + F(\mathbf{B}_t - \mathbf{C}_t)$, where $F \in [0,2]$.

(iii) With a certain probability, perform a crossover between $\mathbf{X}_t^i$ and $\mathbf{Z}_{t+1}$, i.e. $[\mathbf{Z}_{t+1}]_i = [\mathbf{X}_t]_i$, for chosen indices i.

(iv) If $f(\mathbf{Z}_{t+1}) < f(\mathbf{X}_t^i)$, replace $\mathbf{X}_t^i$ by $\mathbf{Z}_{t+1}$ in the population (f is the objective function to minimize).

The crossover relies on the coordinate system; therefore—unless the crossover is not used—DE is not rotational-invariant. Additionally, DE suffers from stagnation. This problem is discussed in [63].

Evolution Strategies

Evolution strategies (ESs) [48, 20, 19] are comparison-based evolutionary algorithms for continuous black-box optimization problems that were first introduced in [81]. One iteration of a general ES can be summarized in the following steps:

(i) First, λ i.i.d. random vectors $\mathbf{U}_{t+1}^i$ are sampled from a multivariate normal distribution of mean $\mathbf{0} \in \mathbb{R}^n$ and covariance matrix $\mathbf{C}_t \in \mathbb{R}^{n \times n}$, and we denote $\mathbf{U}_{t+1}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_t)$.

(ii) The vectors $\mathbf{U}_{t+1}^i$ are then used to create λ candidate solutions $\mathbf{X}_{t+1}^i$ by applying a mutation to the current solution $\mathbf{X}_t$ according to

$$\mathbf{X}_{t+1}^i = \mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^i, \; i = 1, \cdots, \lambda \quad , \tag{3.8}$$

where $\sigma_t > 0$ is called the *step-size* and determines the length of the step taken away from $\mathbf{X}_t$. Notice that (3.8) can also be interpreted as sampling λ candidate solutions $\mathbf{X}_{t+1}^i$ from a normal distribution of mean $\mathbf{X}_t$ and covariance matrix $\sigma_t^2 \mathbf{C}_t$ (i.e. $\mathbf{X}_{t+1}^i \sim \mathcal{N}(\mathbf{X}_t, \sigma_t^2 \mathbf{C}_t)$), where σ_t determines the "width" of the distribution and $\mathbf{C}_t$ determines its "shape". By analogy with the general definition of a comparison-based randomized algorithm presented in Subsection 3.3.3, the probability distribution $p_{\mathbf{U}}$ in this case is defined as

$$p_{\mathbf{U}}(\boldsymbol{u}^1, \cdots, \boldsymbol{u}^\lambda) = p_{\mathcal{N}}(\boldsymbol{u}^1) \cdots p_{\mathcal{N}}(\boldsymbol{u}^\lambda) \quad , \tag{3.9}$$

where $p_{\mathcal{N}}$ is the probability distribution of the multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{C}_t)$. Additionally, the solution function *Sol* (see (3.3)) is defined as

$$Sol((\mathbf{x}, \sigma), \boldsymbol{u}) = \mathbf{x} + \sigma \boldsymbol{u} \quad , \tag{3.10}$$

for an ES whose state is given by $(\mathbf{X}_t, \sigma_t)$.

(iii) Depending on the selection scheme, μ individuals (parents) are selected according to their fitness to create the new solution $\mathbf{X}_{t+1}$ (a) from the new population $\{\mathbf{X}_{t+1}^1, \cdots, \mathbf{X}_{t+1}^\lambda\}$ in the case of non-elitist selection or (b) from the new population and the μ best individuals of the previous population in the case of elitist selection. We denote

- (μ, λ)-ES: an ES using non-elitist (or *comma*) selection.
- $(\mu + \lambda)$-ES: an ES using elitist (or *plus*) selection.

(iv) The new solution $\mathbf{X}_{t+1}$ is obtained via *recombination*, that is, by computing a weighted sum of the μ selected individuals. We distinguish between *weighted* recombination—denoted $(\mu/\mu_W \dot{+} \lambda)$— where the weights are different and *intermediate* recombination—denoted $(\mu/\mu_I \dot{+} \lambda)$—where all the weights are equal [48]. Equation (3.11) below gives the recombination formula in the case of a $(\mu/\mu_W, \lambda)$-ES.

$$\mathbf{X}_{t+1} = \sum_{i=1}^{\mu} w_i \mathbf{X}_{t+1}^{\varsigma(i)} = \mathbf{X}_t + \sigma_t \sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{\varsigma(i)} \quad , \tag{3.11}$$

where we used (3.8) and where ς is the permutation that contains the indices of ranked offspring defined in (3.4). The weights $0 \leq w_i < 1$, $i = 1, \cdots, \mu$, satisfy $\sum_{i=1}^{\mu} w_i = 1$.

An important aspect of ESs is the control of the parameters σ_t and $\mathbf{C}_t$ of the mutation: an ES should choose the step-size σ_t and the covariance matrix $\mathbf{C}_t$ depending on the "context", that is, on the currently explored region of the search space, in order to converge to an optimal solution.

Step-Size Control

It is well-established that the convergence of an ES directly depends on how it controls the step-size. Moreover, the step-size control influences to a large extent the rate at which an ES approaches the optimum. It has been shown that the optimal convergence rate on the sphere function is achieved by choosing a step-size proportional to the distance to the optimum at each iteration [13, Theorem 2]. In the following, we present four step-size adaptation

mechanisms: one-fifth ($1/5$th) success rule, cumulative step-size adaptation, two-point step-size adaptation, and median success rule. The last three mechanisms are implemented in the state-of-the-art ES, the covariance matrix adaptation evolution strategy (CMA-ES) [53]. These step-size adaptation mechanisms are all based on the idea that the smaller the step-size, the higher the probability of sampling "good solutions". In the algorithms we present in Chapters 5 and 6, either of these mechanisms is implemented.

One-fifth success rule The one-fifth (also $1/5$th) success rule is one of the earliest step-size adaptation mechanisms. It was introduced for the first time in [81] for a $(1+1)$-ES. The idea is to maintain a probability of $1/5$ to sample a successful offspring, that is, an offspring whose fitness is better than the one of the current solution. To that end, σ_t is multiplied by $2^{1/n}$ in case of a success and by $2^{-1/(4n)}$ otherwise. Consequently, if one successful solution is sampled every 5 iterations, the step-size remains unchanged. The transition function of a $(1+1)$-ES with $1/5$th success rule is given by

$$\mathscr{G}_{1/5th}((\mathbf{x},\sigma),\mathbf{y}) = \begin{pmatrix} \mathbf{x}+\sigma[\mathbf{y}]_1 \\ \sigma 2^{-\frac{1}{4n}+\frac{5}{4n}\mathbf{1}_{\{[\mathbf{y}]_1\neq\mathbf{0}\}}} \end{pmatrix} ,$$

where we assume the state $\mathbf{s}_t = (\mathbf{X}_t, \sigma_t)$ and where $[\mathbf{y}]_1$ is the the first element of the vector $\mathbf{y}$ (more generally, $[\mathbf{y}]_i$ denotes the ith element of some vector $\mathbf{y}$).

Cumulative step-size adaptation Cumulative step-size adaptation (CSA) [53] is the default step-size adaptation mechanism in the state-of-the-art ES, CMA-ES. The (normalized) steps taken by the algorithm in the search space are recorded by computing the so-called *evolution path* $\boldsymbol{p}_t^\sigma$ according to

$$\begin{aligned} \boldsymbol{p}_{t+1}^\sigma &= (1-c_\sigma)\boldsymbol{p}_t^\sigma + \sqrt{c_\sigma(2-c_\sigma)/\sum_{i=1}^{\mu} w_i^2}\,\mathbf{C}_t^{-1/2}\left(\frac{\mathbf{X}_{t+1}-\mathbf{X}_t}{\sigma_t}\right) \\ &= (1-c_\sigma)\boldsymbol{p}_t^\sigma + \sqrt{c_\sigma(2-c_\sigma)/\sum_{i=1}^{\mu} w_i^2}\,\mathbf{C}_t^{-1/2}\sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{\varsigma(i)} , \end{aligned} \tag{3.12}$$

for a $(\mu/\mu_W,\lambda)$-ES, where $0 < c_\sigma \leq 1$, $\boldsymbol{p}_0^\sigma = \mathbf{0}$, and ς is defined in (3.4). The coefficient $\sqrt{c_\sigma(2-c_\sigma)/\sum_{i=1}^{\mu} w_i^2}$ is chosen such that if $\boldsymbol{p}_t^\sigma \sim \mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})$ and if the μ best individuals are selected randomly, then $\boldsymbol{p}_{t+1}^\sigma \sim \mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})$ [48, 50]. The length of the evolution path $\|\boldsymbol{p}_{t+1}^\sigma\|$ is then investigated: if $\|\boldsymbol{p}_{t+1}^\sigma\|$ is "too large", this means that many successive steps are made in the same direction and, therefore, that the progress is too slow. Consequently, σ_t is increased. In contrast, if $\|\boldsymbol{p}_{t+1}^\sigma\|$ is "too small", this suggests that the steps taken by the

algorithm are in opposite directions and that the algorithm may be overshooting the optimum. In this case, σ_t is reduced. This is formally translated by the following adaptation rule

$$\sigma_{t+1} = \sigma_t \exp^{\frac{c_\sigma}{d_\sigma}} \left(\frac{\|\boldsymbol{p}^\sigma_{t+1}\|}{E\|\mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})\|} - 1 \right) \quad , \tag{3.13}$$

where $\|\boldsymbol{p}^\sigma_{t+1}\|$ is compared to the expected length of a multivariate standard normal vector in practice: if the ratio $\frac{\|\boldsymbol{p}^\sigma_{t+1}\|}{E\|\mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})\|} > 1$, σ_t is increased; if $\frac{\|\boldsymbol{p}^\sigma_{t+1}\|}{E\|\mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})\|} < 1$, σ_t is decreased; if $\frac{\|\boldsymbol{p}^\sigma_{t+1}\|}{E\|\mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})\|} = 1$, σ_t is unchanged. The positive constant $d_\sigma \geq 1$ is a damping factor whose role is to attenuate the variations of σ_t. The transition function of an ES with CSA and with a fixed covariance matrix $\mathbf{C}_t = \mathbf{I}_{n\times n}$ is given by

$$\mathscr{G}_{\text{CSA}}((\mathbf{x}, \boldsymbol{p}, \sigma), \mathbf{y}) = \begin{pmatrix} \mathbf{x} + \sigma \sum_{i=1}^{\mu} [\mathbf{y}]_i \\ (1-c_\sigma)\boldsymbol{p} + \sqrt{c_\sigma(2-c_\sigma)/\sum_{i=1}^{\mu} w_i^2} \sum_{i=1}^{\mu} w_i [\mathbf{y}]_i = \boldsymbol{p}' \\ \sigma \exp^{\frac{c_\sigma}{d_\sigma}} \left(\frac{\|\boldsymbol{p}'\|}{E\|\mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})\|} - 1 \right) \end{pmatrix} \quad ,$$

where we consider that $\mathbf{s}_t = (\mathbf{X}_t, \boldsymbol{p}^\sigma_t, \sigma_t)$.

Two-point step-size adaptation Two-point step-size adaptation (TPA) is a relatively new step-size adaptation mechanism. We present here the implementation in [51], which is based on [49, 82]. First, two offspring $\mathbf{X}^1_{t+1}$ and $\mathbf{X}^2_{t+1}$ are sampled as a mirrored pair along the line connecting the current solution $\mathbf{X}_t$ to the previous one $\mathbf{X}_{t-1}$, and symmetric to $\mathbf{X}_t$ as follows

$$\mathbf{X}^{1,2}_{t+1} = \mathbf{X}_t \pm \sigma_t \times \|\mathbf{U}^{1,2}_{t+1}\| \times \frac{\mathbf{X}_t - \mathbf{X}_{t-1}}{\|\mathbf{X}_t - \mathbf{X}_{t-1}\|} \quad , \tag{3.14}$$

where $\mathbf{U}^{1,2}_{t+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$. Notice that $\mathbf{X}^2_{t+1}$ lies between $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$. If $\mathbf{X}^1_{t+1}$ is fitter than $\mathbf{X}^2_{t+1}$, one can assume that better solutions are available in the direction of the last solution shift and the step-size is increased. Otherwise, it is decreased. Formally, this update is given by the equations below:

$$\begin{aligned} s_{t+1} &= (1-c_\sigma)s_t + c_\sigma \frac{\text{rank}(\mathbf{X}^2_{t+1}) - \text{rank}(\mathbf{X}^1_{t+1})}{\lambda - 1} \quad , \\ \sigma_{t+1} &= \sigma_t \exp\left(\frac{s_{t+1}}{d_\sigma}\right) \quad , \end{aligned} \tag{3.15}$$

where the function $\text{rank}(\mathbf{X}^i_{t+1})$ returns the rank of the individual $\mathbf{X}^i_{t+1}$ in the population, $s_0 = 0$, $0 < c_\sigma \leq 1$, and $d_\sigma \geq 1$ is a damping factor to moderate the changes of σ_t. The $\lambda - 2$ remaining offspring are sampled as in (3.8). In the case of TPA, the solution function differs

slightly from the general solution function *Sol* defined in (3.10) since the first two offspring are sampled differently from the remaining $\lambda - 2$ offspring. Therefore, this algorithm is not captured by the general definition we present in Subsection 3.3.3.

Median success rule Another recent step-size adaptation mechanism is the so-called median success rule (MSR) [1]. It generalizes the $1/5$th success rule introduced previously to the case of non-elitist ESs by redefining the notion of "success" as the median individual (in terms of f-value), denoted $\mathbf{X}_{t+1}^{\boldsymbol{m}(\lambda)}$, being better than the jth best individual in the previous population, denoted $\mathbf{X}_t^{j:\lambda}$, where j is fixed. In practice, we set j to the 30th percentile; for this value, the median success probability is roughly $1/2$ on the sphere function with optimal step-size[4] [1]. Similarly to the $1/5$th success rule, the step-size is increased in case of success and decreased otherwise in order to increase the probability of sampling successful offspring. The exact adaptation rule is given by equations below.

$$z_t = \frac{2}{\lambda}\left(K_{\text{succ}} - \frac{2}{\lambda}\right) \quad , \tag{3.16}$$

$$q_{t+1} = (1-c_\sigma)q_t + c_\sigma z_t \quad , \tag{3.17}$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{q_{t+1}}{d_\sigma}\right) \quad . \tag{3.18}$$

The success in measured by computing z_t in (3.16), where K_{succ} is the number of offspring better than $\mathbf{X}_t^{j:\lambda}$. Notice that $K_{\text{succ}} \geq 2/\lambda$ is equivalent to $\mathbf{X}_{t+1}^{\boldsymbol{m}(\lambda)}$ being better than $\mathbf{X}_t^{j:\lambda}$. It results that $z_t \geq 0$ if and only if $\mathbf{X}_{t+1}^{\boldsymbol{m}(\lambda)}$ is successful. In (3.17), z_t is cumulated in q_{t+1}, where $0 < c_\sigma \leq 1$. The step-size σ_t is updated in (3.18): it is increased if $q_{t+1} > 0$ (success) and decreased otherwise. The transition function for MSR is given by the following equation:

$$\mathscr{G}^f_{\text{MSR}}((\mathbf{x},q,\boldsymbol{v},\sigma),\mathbf{y}) = \begin{pmatrix} \mathbf{x} + \sigma \sum_{i=1}^{\mu} [\mathbf{y}]_i \\ (1-c_\sigma)q + c_\sigma \frac{2}{\lambda}\left(\sum_{i=1}^{\lambda} \mathbf{1}_{\{f(\mathbf{x}+\sigma[\mathbf{y}]_i) \leq f(\boldsymbol{v})\}} - \frac{2}{\lambda}\right) = q' \\ \boldsymbol{v} + \sigma \mathbf{y}^j \\ \sigma \exp(q') \end{pmatrix} ,$$

where we consider that the state $\mathbf{s}_t = (\mathbf{X}_t, q_t, \mathbf{X}_t^{j:\lambda}, \sigma_t)$. Note that for MSR, the objective function f is used in the transition function since computing the success implies comparing f-values of the current population to the f-value of the jth best individual of the previous iteration. However, these f-values are only used through comparison.

[4]The optimal step-size adaptation strategy on the sphere function is to choose a step-size proportional to the distance to the optimum at each iteration. Notice however that this is an artificial algorithm since the optimum is usually unknown.

Covariance Matrix Adaptation

As mentioned previously in this section, the covariance matrix $\mathbf{C}_t$ determines the shape of the sampling distribution. The rationale behind the adaptation of the covariance matrix is to maximize the likelihood of "good" solutions. This is particularly relevant on ill-conditioned and non-separable functions. In [53], the first covariance matrix adaptation evolution strategy (CMA-ES) is presented. Similarly to CSA, CMA-ES uses an evolution path $\boldsymbol{p}_t$ to adapt the covariance matrix as follows:

$$\begin{aligned}\boldsymbol{p}_{t+1} &= (1-c_c)\boldsymbol{p}_t + \sqrt{c_c(2-c_c)/\sum_{i=1}^{\mu} w_i^2}\left(\frac{\mathbf{X}_{t+1}-\mathbf{X}_t}{\sigma_t}\right) \\ &= (1-c_c)\boldsymbol{p}_t + \sqrt{c_c(2-c_c)/\sum_{i=1}^{\mu} w_i^2}\sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{\varsigma(i)} \ , \end{aligned} \tag{3.19}$$

where $0 < c_c \leq 1$. Compared to $\boldsymbol{p}_{t+1}^{\sigma}$ (3.12), the information provided by $\mathbf{C}_t$ is used in $\boldsymbol{p}_{t+1}$. The covariance matrix is then updated as follows:

$$\mathbf{C}_{t+1} = (1-c_1-c_\mu)\mathbf{C}_t + \underbrace{c_1 \boldsymbol{p}_{t+1}\boldsymbol{p}_{t+1}^{\mathsf{T}}}_{\text{rank-one update}} + \underbrace{c_\mu \sum_{i=1}^{\mu} w_i \left(\frac{\mathbf{X}_{t+1}^{i:\lambda}-\mathbf{X}_t}{\sigma_t}\right)\left(\frac{\mathbf{X}_{t+1}^{i:\lambda}-\mathbf{X}_t}{\sigma_t}\right)^{\mathsf{T}}}_{\text{rank-}\mu\text{ update}} \ , \tag{3.20}$$

where $0 < c_1 \leq 1$ and $0 < c_\mu \leq 1$. This update combines the so-called *rank-one* and *rank-μ* updates: the former is associated to c_1 and reshapes $\mathbf{C}_t$ in the direction of $\boldsymbol{p}_t$, and the latter is associated to c_μ and reshapes $\mathbf{C}_t$ in the direction of the best offspring of the current population.

CMA-ES has become the state-of-the-art ES thanks to its efficiency on a wide range of optimization problems including ill-conditioned and non-separable problems [16].

3.4 Invariance

Invariance is an important property in optimization because it reflects the robustness of an algorithm. Typically, when an algorithm is invariant, the performance observed on a particular problem can be generalized to an entire class of problems [15]. We distinguish between invariance to transformations of the objective function and invariance to transformations of the search space. By definition, comparison-based algorithms are invariant to strictly monotonic transformations of the objective function f. Indeed, if $g : \mathbb{R} \to \mathbb{R}$ is a strictly

increasing function, i.e. for all $x, y \in \mathbb{R}$ such that $x < y$, $g(x) < g(y)$, then

$$Ord(f(\mathbf{x}_i)_{i=1,\cdots,p}) = Ord(g \circ f(\mathbf{x}_i)_{i=1,\cdots,p}) \quad ,$$

that is, the ranking of p candidate solutions $\mathbf{x}_i$ according to their f-values and according to their $g \circ f$-values is the same.

As for invariance to transformations of the search space, a general definition that includes *translation*-invariance, *scale*-invariance, *affine*-invariance, and *rotational*-invariance is presented in [11], as well as a proof of affine-invariance of CMA-ES with a modified version of CSA. We will focus here on translation-invariance and scale-invariance as they are key for proving the convergence of many comparison-based randomized adaptive algorithms [15]. We start by giving the definition of group homomorphism and introducing some notations that will help us define translation-invariance and scale-invariance in the case of a randomized algorithm for unconstrained optimization. These definitions are taken from [15] and will be extended to the case of constrained optimization in Chapter 6.

Definition 1. Let $(G_1, .)$ and $(G_2, *)$ be two groups. The mapping $\Phi : G_1 \to G_2$ is a group homomorphism if for all $x, y \in G_1$, $\Phi(x.y) = \Phi(x) * \Phi(y)$.

We denote by $\mathscr{S}(\Omega)$ the set of all bijective transformations from a set Ω to itself and by $\text{Homo}((\mathbb{R}^n, +), (\mathscr{S}(\Omega), \circ))$ (respectively $\text{Homo}((\mathbb{R}^+_>, .), (\mathscr{S}(\Omega), \circ))$) the set of group homomorphisms from $(\mathbb{R}^n, +)$ (respectively from $(\mathbb{R}^+_>, .)$) to $(\mathscr{S}(\Omega), \circ)$.

If an algorithm is translation-invariant, this means that it performs similarly on $\mathbf{x} \mapsto f(\mathbf{x})$ and $\mathbf{x} \mapsto f(\mathbf{x} - \mathbf{x}_0)$ for any $\mathbf{x}_0$ and, therefore, that the choice of the initial solution does not affect the performance of the algorithm. The following definition formally defines translation-invariance. It states that translation-invariance holds if there exists a state space transformation—here a group homomorphism from $(\mathbb{R}^n, +)$ to $\mathscr{S}(\Omega)$—for which optimizing $\mathbf{x} \mapsto f(\mathbf{x})$ in the original state space is equivalent to optimizing $\mathbf{x} \mapsto f(\mathbf{x} - \mathbf{x}_0)$ in the transformed state space.

Definition 2. A randomized algorithm with transition function $\mathscr{F}^f : \Omega \times (\mathbb{R}^n)^\lambda \to \Omega$, where f is the objective function to minimize, is translation-invariant if there exists a group homomorphism $\Phi \in \text{Homo}((\mathbb{R}^n, +), (\mathscr{S}(\Omega), \circ))$ such that for any objective function f, for any $\mathbf{x}_0 \in \mathbb{R}^n$, for any state $\mathbf{s} \in \Omega$, and for any $\boldsymbol{u} \in (\mathbb{R}^n)^\lambda$,

$$\mathscr{F}^{f(\mathbf{x})}(\mathbf{s}, \boldsymbol{u}) = \Phi(-\mathbf{x}_0)\left(\mathscr{F}^{f(\mathbf{x}-\mathbf{x}_0)}(\Phi(\mathbf{x}_0)(\mathbf{s}), \boldsymbol{u})\right) \quad .$$

Analogously to translation-invariance, scale-invariance holds if there exists a search space transformation such that minimizing $\mathbf{x} \mapsto f(\mathbf{x})$ in the original state space is equivalent to

minimizing $\mathbf{x} \mapsto f(\alpha\mathbf{x})$ for any $\alpha > 0$ in the transformed search space. The formal definition is given below.

Definition 3. A randomized algorithm with transition function $\mathscr{F}^f : \Omega \times (\mathbb{R}^n)^\lambda \to \Omega$, where f is the objective function to minimize, is scale-invariant if there exists a group homomorphism $\Phi \in \mathrm{Homo}((\mathbb{R}_>^+, .), (\mathscr{S}(\Omega), \circ))$ such that for any objective function f, for any $\alpha > 0$, for any state $\mathbf{s} \in \Omega$, and for any $\boldsymbol{u} \in (\mathbb{R}^n)^\lambda$,

$$\mathscr{F}^{f(\mathbf{x})}(\mathbf{s}, \boldsymbol{u}) = \Phi(1/\alpha)\left(\mathscr{F}^{f(\alpha\mathbf{x})}(\Phi(\alpha)(\mathbf{s}), \boldsymbol{u})\right) \quad .$$

In [15, 14], the authors prove translation-invariance and scale-invariance of a general comparison-based step-size adaptive randomized algorithm with state $\mathbf{s}_t = (\mathbf{X}_t, \sigma_t)$, given the following sufficient conditions are satisfied by its transition function $\mathscr{G}((\mathbf{x}, \sigma), \mathbf{y}) = (\mathscr{G}_\mathbf{x}((\mathbf{x}, \sigma), \mathbf{y}), \mathscr{G}_\sigma(\sigma, \mathbf{y}))$:

(i) for all $\mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^n$, for all $\sigma > 0$, for all $\mathbf{y} \in (\mathbb{R}^n)^\lambda$

$$\mathscr{G}_\mathbf{x}((\mathbf{x} + \mathbf{x}_0, \sigma), \mathbf{y}) = \mathscr{G}_\mathbf{x}((\mathbf{x}, \sigma), \mathbf{y}) + \mathbf{x}_0 \quad , \tag{3.21}$$

(ii) for all $\mathbf{x} \in \mathbb{R}^n$, for all $\alpha, \sigma > 0$, for all $\mathbf{y} \in (\mathbb{R}^n)^\lambda$

$$\mathscr{G}_\mathbf{x}((\mathbf{x}, \sigma), \mathbf{y}) = \alpha\mathscr{G}_\mathbf{x}\left(\left(\frac{\mathbf{x}}{\alpha}, \frac{\sigma}{\alpha}\right), \mathbf{y}\right) \quad , \tag{3.22}$$

(iii) for all $\alpha, \sigma > 0$, for all $\mathbf{y} \in (\mathbb{R}^n)^\lambda$

$$\mathscr{G}_\sigma(\sigma, \mathbf{y}) = \alpha\mathscr{G}_\sigma\left(\frac{\sigma}{\alpha}, \mathbf{y}\right) \quad , \tag{3.23}$$

where $\mathscr{G}_\mathbf{x}$ (respectively $\mathscr{G}_\sigma$) is the update function for the current solution (respectively the step-size). Conditions (i) and (ii) are naturally satisfied for evolution strategies (see (3.11)). Condition (iii), on the other hand, is satisfied for evolution strategies with *multiplicative* step-size adaptation rules similar to the ones presented above ($1/5$th success rule, CSA, TPA, and MSR).

As mentioned previously in this section, translation-invariance and scale-invariance are key elements in proving linear convergence of comparison-based adaptive randomized algorithms for unconstrained optimization via Markov chain theory [15, 14]. As shown in [15, Proposition 4.1], if conditions (3.21), (3.22), and (3.23) hold, then the sequence $(\mathbf{Y}_t)_{t\in\mathbb{N}}$ of random variables $\mathbf{Y}_t = \frac{\mathbf{X}_t - \mathbf{x}_{\mathrm{opt}}}{\sigma_t}$ is a homogeneous Markov chain on the class of scaling-

invariant objective functions, where $\mathbf{x}_{\text{opt}}$ is the optimal solution. This result is exploited to deduce linear convergence of the algorithm, as we will illustrate in Subsection 3.5.2.

3.5 Evaluating the Performance

In this section, we discuss various aspects related to the evaluation of the performance of algorithms for black-box continuous optimization.

In real-world applications, the evaluation of an objective function can be computationally costly. Therefore, a black-box optimization algorithm is usually allocated a *budget* consisting in a fixed number of function evaluations to try to find the optimum. A natural way to compare two algorithms would be to compare the f-values of their respective solutions. This measure however is not quantitative. Indeed, having a function value two times smaller than another, for instance, does not give much information on the actual performance of the algorithms. Instead, the so-called *average runtime* (aRT), is commonly used in practice, particularly in one of the best-known benchmarking platforms for continuous optimization, the COCO (comparing continuous optimizers) platform [52]. The convergence rate of an algorithm, i.e. the speed at which it reaches the optimum, can also be investigated. We focus on *linear convergence* in this work since it is the fastest possible convergence for a comparison-based algorithm.

In the following, we define the aRT and discuss linear convergence and how it can be investigated using the Markov chain theory.

3.5.1 Average Runtime

The average runtime (aRT) was originally introduced in [80]. Let us consider a randomized algorithm minimizing a function $f : \mathscr{X} \subseteq \mathbb{R}^n \to \mathbb{R}$. Given a budget of b function evaluations and a target f-value, f_{target}, we run different instances of the algorithm on f. A run is said to be "successful" if the target value f_{target} is reached within the given budget b. The aRT is an estimate of the average runtime RT (in terms of the number function evaluations) needed by an algorithm to reach a given target value and defined as [12]

$$E(\text{RT}) = E(\text{RT}^s) + \frac{1-p_s}{p_s} E(\text{RT}^{us}) \quad ,$$

where RT^s (respectively RT^{us}) is the runtime of a successful (respectively unsuccessful) run of the algorithm and where p_s is the probability of a successful run. The aRT is defined as

$$\begin{aligned}\mathrm{aRT} &= \frac{1}{n_s}\sum_i \mathrm{RT}_i^s + \frac{1-p_s}{p_s}\frac{1}{n_{us}}\sum_j \mathrm{RT}_j^{us} \\ &= \frac{\sum_i \mathrm{RT}_i^s + \sum_j \mathrm{RT}_j^{us}}{n_s} = \frac{\#\mathrm{FEs}}{n_s} \quad ,\end{aligned}$$

where n_s (respectively n_{us}) is the number of successful (unsuccessful) runs, p_s here is the ratio of successful runs, and #FEs is the number of total function evaluations over all runs [52].

The aRT allows a quantitative comparison of algorithms and is used as a performance measure in the COCO benchmarking platform [52], which provides different testbeds for evaluating continuous optimization algorithms (a noiseless testbed, a noisy testbed, and a bi-objective testbed) and interfaces to use these testbeds with different programming languages. COCO also provides a tool for processing and visualizing the data related to an algorithm automatically. The BBOB noiseless testbed of COCO consists in 24 functions that can be classified into five categories (separable functions, moderate functions, ill-conditioned functions, multi-modal functions, and weakly structured multi-modal functions) depending on their features and the difficulty they reflect for an optimization algorithm. A thorough description of the BBOB testbed is provided in [52].

3.5.2 Linear Convergence

Linear convergence is the fastest possible convergence rate for a comparison-based algorithm. It is a highly desirable property for an algorithm and many randomized algorithms for unconstrained optimization are designed with the purpose of converging linearly on simple optimization problems, such as the sphere function or the linear function[5]. Let us consider the following definition of linear convergence.

Definition 4. A sequence $(\mathbf{X}_t)_{t\in\mathbb{N}}$ of random vectors $\mathbf{X}_t$ is said to converge linearly almost surely (a.s.) to some vector $\mathbf{x}_{\mathrm{opt}}$ if there exists $\mathrm{CR} > 0$ such that

$$\lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\mathbf{X}_t - \mathbf{x}_{\mathrm{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\mathrm{opt}}\|} = -\mathrm{CR} \text{ a.s.}$$

The constant CR gives the convergence rate.

[5] On the linear function, linear *divergence* is desirable.

Informally, the previous definition states that linear convergence holds if the distance to $\mathbf{x}_{\text{opt}}$, $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$, decreases linearly in log-scale. Some of the most efficient randomized algorithms, such as CMA-ES, are empirically observed to converge linearly on a wide range of problems . In constrained optimization, the work of [5] was the first to explicitly consider linear convergence: the authors presented a $(1+1)$-ES for constraint optimization for the case of one inequality constraint. Their algorithm was empirically observed to converge linearly on two linearly constrained convex quadratic functions (the sphere function and a moderately ill-conditioned ellipsoid function).

In the case of unconstrained optimization, linear convergence can be proven for randomized algorithms with proper invariance properties using tools form the Markov chain theory [10, 15, 21]. For the sake of illustration, let us consider a randomized adaptive algorithm with state variables $(\mathbf{X}_t, \sigma_t)$ (the current estimate of the optimum and the step-size respectively) minimizing a sphere function $\mathbf{x} \mapsto f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\intercal\mathbf{x}$ with optimum in zero without loss of generality. The sphere function is scaling-invariant[6]; therefore, if translation-invariance and scale-invariance hold (see sufficient conditions (3.21), (3.22), and (3.23)), the sequence $(\mathbf{Y}_t)_{t\in\mathbb{N}}$, with $\mathbf{Y}_t = \mathbf{X}_t/\sigma_t$, is a homogeneous Markov chain defined independently of $(\mathbf{X}_t, \sigma_t)$, given $\mathbf{Y}_0 = \mathbf{X}_0/\sigma_0$, as

$$\mathbf{Y}_{t+1} = \frac{\mathscr{G}_{\mathbf{x}}((\mathbf{Y}_t, 1), \varsigma * \mathbf{U}_{t+1})}{\mathscr{G}_{\sigma}(1, \varsigma * \mathbf{U}_{t+1})} \quad ,$$

where $\mathbf{U}_{t+1} = [\mathbf{U}_{t+1}^1, \ldots, \mathbf{U}_{t+1}^\lambda]$, $\mathbf{U}_{t+1}^i$, $i = 1, \cdots, \lambda$, are i.i.d. random vectors and $\varsigma = Ord(f(\mathbf{Y}_t + \mathbf{U}_{t+1}^i)_{i=1,\cdots,\lambda})$. Considering the previous definition of linear convergence, we can express $\frac{1}{t}\ln\frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|}$ ($\mathbf{x}_{\text{opt}} = \mathbf{0}$) as a function of the Markov chain $\mathbf{Y}_t$ as in the following:

$$\begin{aligned}\frac{1}{t}\ln\frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|} = \frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{X}_{k+1}\|}{\|\mathbf{X}_k\|} &= \frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{X}_{k+1}\|}{\|\mathbf{X}_k\|}\frac{\sigma_k\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1})}{\sigma_{k+1}} \\ &= \frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{Y}_{k+1}\|}{\|\mathbf{Y}_k\|}\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1}) \quad , \end{aligned} \tag{3.24}$$

where we successively used the property of the logarithm then artificially introduced $\sigma_{k+1} = \sigma_k\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1})$ and used $\mathbf{Y}_k = \mathbf{X}_k/\sigma_k$ and $\mathbf{Y}_{k+1} = \mathbf{X}_{k+1}/\sigma_{k+1}$. If $(\mathbf{Y}_t)_{t\in\mathbb{N}}$ satisfies sufficient stability conditions (see for instance Theorem 1), then a LLN for Markov

[6] We distinguish between scaling-invariance (of a function) and scale-invariance (of an algorithm). A function $f : \mathbb{R}^n \to \mathbb{R}$ is scaling-invariant with respect to $\mathbf{x}^* \in \mathbb{R}^n$ [15, Definition 3.1] if for all $\alpha > 0$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $f(\mathbf{x}^* + \mathbf{x}) \leq f(\mathbf{x}^* + \mathbf{y}) \Leftrightarrow f(\mathbf{x}^* + \alpha\mathbf{x}) \leq f(\mathbf{x}^* + \alpha\mathbf{y})$.

chains can be applied to the right-hand side of (3.24). It follows that

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|}=\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{Y}_{k+1}\|}{\|\mathbf{Y}_k\|}\mathscr{G}_\sigma(1,\varsigma*\mathbf{U}_{k+1})$$
$$=\int\ln\|\mathbf{y}\|\pi(d\mathbf{y})-\int\ln\|\mathbf{y}\|\pi(d\mathbf{y})+\underbrace{\int E_{\mathbf{U}\sim p_\mathbf{U}}(\ln(\mathscr{G}_\sigma(1,\varsigma*\mathbf{U}))|\mathbf{Y}_t=\mathbf{y})\pi(d\mathbf{y})}_{-\mathrm{CR}},$$

where π is the invariant probability measure of the Markov chain $(\mathbf{Y}_t)_{t\in\mathbb{N}}$ and $p_\mathbf{U}$ is the probability distribution of $\mathbf{U}_{t+1}$ (see (3.9)). Therefore, assuming the Markov chain $(\mathbf{Y}_t)_{t\in\mathbb{N}}$ is stable, the randomized algorithm described here will converge to $\mathbf{x}_{\mathrm{opt}}=\mathbf{0}$ (on any scaling-invariant function) at a speed given by the expected log step-size change with respect to the stationary distribution of $\mathbf{Y}_t$.

As illustrated with the previous example, the analysis of linear convergence with a Markov chain approach consists in two steps:

(i) First, identify a class of functions on which a homogeneous Markov chain can be constructed from the state variables of the algorithm at hand and such that its stability leads to linear convergence of the algorithm.

(ii) Then, prove the stability of the constructed Markov chain.

The second step is the most difficult in practice and is outside the scope of this work. For further reading, we point to [10, 14, 26] where stability is proven for self-adaptive algorithms, for the $(1+1)$-ES with one-fifth success rule, and for a $(1,\lambda)$-ES with CSA respectively, in the case of unconstrained optimization. In [27], stability is proven for a $(1,\lambda)$-ES on the linearly constrained linear function and in [28], the authors study a $(1,\lambda)$-ES with a general sampling distribution on the linearly constrained linear function and give sufficient conditions on the sampling distribution for positivity, Harris-recurrence, and ergodicity of the studied Markov chain. The first step of the analysis, which was achieved for comparison-based randomized step-size adaptive algorithms on scaling-invariant functions in [15], is the focus of this work. In particular, we generalize the approach presented above to the case of constrained optimization where the constraints are linear and identify a Markov chain for (i) the $(1+1)$-ES with one-fifth success rule in the case of a single linear constraint and (ii) a general comparison-based step-size adaptive algorithm in the case of m linear constraints. Both algorithms handle the constraints with an *augmented Lagrangian* approach and are presented in Chapter 6.

Chapter 4

Constrained Optimization

Numerical optimization problems encountered in practice are often *constrained*, that is, given an objective function $f : \mathscr{X} \subseteq \mathbb{R}^n \to \mathbb{R}$ to minimize, a solution must belong to a *feasible set* $\mathscr{X}_F \subseteq \mathscr{X}$ determined by a set of equality and inequality constraints. In its most general form, a constrained optimization problem is formulated as

$$\begin{aligned} &\min_{\mathbf{x}\in\mathscr{X}} f(\mathbf{x}) \\ &\text{subject to } g_i(\mathbf{x}) = 0, \quad i \in \mathscr{E} \\ &\qquad\qquad g_i(\mathbf{x}) \leq 0, \quad i \in \mathscr{I} \quad , \end{aligned} \tag{4.1}$$

where $g_i : \mathscr{X}_{c_i} \subseteq \mathbb{R}^n \to \mathbb{R}$, $i \in \mathscr{E}$, are the *equality constraints*, $g_i : \mathscr{X}_{g_i} \subseteq \mathbb{R}^n \to \mathbb{R}$, $i \in \mathscr{I}$, are the *inequality constraints*, and $\mathscr{E}$ and $\mathscr{I}$ are two finite sets of indices that satisfy $\mathscr{E} \cap \mathscr{I} = \emptyset$. Problem (4.1) can also be written as

$$\min_{\mathbf{x}\in\mathscr{X}_F} f(\mathbf{x}) \quad ,$$

where $\mathscr{X}_F$ is the feasible set which we formally define later on in this chapter. We refer to points in $\mathscr{X}_F$ as *feasible points* and to points in $\mathscr{X} \setminus \mathscr{X}_F$ as *unfeasible points*.

We talk about (i) *linear programming* when the objective function f and all the constraints g_i are linear functions, (ii) *nonlinear programming* when some of the constraints or the objective function are nonlinear and (iii) *quadratic programming*, as a particular case, when f is a quadratic function of the form $\mathbf{x} \mapsto f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\intercal \mathbf{G}\mathbf{x} + \mathbf{x}^\intercal \mathbf{v}$ (with $\mathbf{G} \in \mathbb{R}^{n\times n}$ a symmetric matrix and $\mathbf{v} \in \mathbb{R}^n$) and the constraints g_i are linear. In a black-box context, no information about the nature of the objective function is available—although the constraints might sometimes be known [3]. For this reason, we focus our attention on algorithms for the more general case of nonlinear programming in the sequel.

This chapter is intended as a general introduction to numerical constrained optimization. In Section 4.1, we recall some fundamental definitions as well as optimality conditions characterizing a solution to the problem in (4.1). In Section 4.2, we review some of the most famous nonlinear programming methods as well as some constraint handling techniques for evolutionary algorithms. We finish with a discussion in Section 4.3.

4.1 Theory of Constrained Optimization

We start this section by some basic yet important definitions. Most of these definitions can be found in [73, 44, 18].

Definition 5 (Feasible Set)**.** The *feasible set* $\mathscr{X}_F$ of points satisfying the constraints in (4.1) is formally defined as

$$\mathscr{X}_F = \{\mathbf{x} \mid g_i(\mathbf{x}) = 0,\ i \in \mathscr{E};\ g_i(\mathbf{x}) \leq 0,\ i \in \mathscr{I}\} \ . \tag{4.2}$$

Definition 6 (Local Optimum)**.** A point $\mathbf{x}^*$ is a *local optimum* of the optimization problem (4.1) if $\mathbf{x}^* \in \mathscr{X}_F$ and there exists a neighborhood $\mathscr{N}$ of $\mathbf{x}^*$ such that for all $\mathbf{x} \in \mathscr{N} \cap \mathscr{X}_F$, $f(\mathbf{x}^*) \leq f(\mathbf{x})$.

Definition 7 (Strict Local Optimum)**.** A point $\mathbf{x}^*$ is a *strict local optimum* of the optimization problem (4.1) if $\mathbf{x}^* \in \mathscr{X}_F$ and there exists a neighborhood $\mathscr{N}$ of $\mathbf{x}^*$ such that for all $\mathbf{x} \in \mathscr{N} \cap \mathscr{X}_F \setminus \{\mathbf{x}^*\}$, $f(\mathbf{x}^*) < f(\mathbf{x})$.

Definition 8 (Global Optimum)**.** A point $\mathbf{x}^*$ is a *global optimum* of the optimization problem (4.1) if for any neighborhood $\mathscr{N}$ of $\mathbf{x}^*$, $\mathbf{x}^*$ is a local optimum.

Definition 9 (Active Set and Active Constraint)**.** We call *active set* at some feasible point $\mathbf{x} \in \mathscr{X}_F$ the set $\mathscr{A}(\mathbf{x})$ of indices of all equality constraints and indices i of inequality constraints satisfying $g_i(\mathbf{x}) = 0$. That is

$$\mathscr{A}(\mathbf{x}) = \mathscr{E} \cup \{i \in \mathscr{I} \mid g_i(\mathbf{x}) = 0\} \ . \tag{4.3}$$

An inequality constraint whose index i is in $\mathscr{A}(\mathbf{x})$ is said to be *active* at $\mathbf{x}$.

The following definition introduces the *linearized feasible direction set* [73] which contains the feasible directions at a feasible point $\mathbf{x}$, obtained by approximating linearly the constraint functions g_i at $\mathbf{x}$.

Definition 10 (Linearized Feasible Direction Set)**.** Assume the constraint functions g_i are differentiable. Given a feasible point $\mathbf{x}$ and the active set $\mathscr{A}(\mathbf{x})$, the set of linearized feasible directions $\mathscr{F}(\mathbf{x})$ is defined as

$$\mathscr{F}(\mathbf{x}) = \{\mathbf{d} \mid \nabla_{\mathbf{x}} g_i(\mathbf{x})^{[1]}\mathbf{d} = 0,\ i \in \mathscr{E};\ \nabla_{\mathbf{x}} g_i(\mathbf{x})\mathbf{d} \leq 0,\ i \in \mathscr{I}\} \ .$$

Assuming the optimization problem in (4.1) admits at least a solution, optimality conditions describe the relation between the objective function f and the constraint functions g_i at a local minimum $\mathbf{x}^*$ of (4.1). They are generally expressed as *necessary* conditions on $\mathbf{x}^*$ and assume the functions f and g_i to be *smooth*, that is, their second derivatives exist and are continuous [73]. They also assume some regularity conditions—or *constraint qualifications*—are satisfied by the constraint functions at $\mathbf{x}^*$. Two main constraint qualifications are used in the mathematical programming literature, namely the *linear independence constraint qualification* (LICQ) and the *Mangasarian-Fromovitz constraint qualification* (MFCQ), which are defined in Definitions 11 and 12 below.

Definition 11 (LICQ)**.** The *linear independence constraint qualification* (LICQ) holds at a point $\bar{\mathbf{x}} \in \mathscr{X}_F$ if the gradients of all active constraints at $\bar{\mathbf{x}}$, $\nabla_{\mathbf{x}} g_i(\bar{\mathbf{x}})$, $i \in \mathscr{A}(\bar{\mathbf{x}})$, are linearly independent.

Definition 12 (MFCQ)**.** The *Mangasarian-Fromovitz constraint qualification* (MFCQ) holds at a point $\bar{\mathbf{x}} \in \mathscr{X}_F$ if the gradients of all equality constraints, $\nabla_{\mathbf{x}} g_i(\bar{\mathbf{x}})$, $i \in \mathscr{E}$, are linearly independent and if there exists a vector $\mathbf{v} \in \mathbb{R}^n$ such that

$$\begin{aligned} \nabla_{\mathbf{x}} g_i(\bar{\mathbf{x}})\mathbf{v} &< 0,\ i \in \mathscr{A}(\bar{\mathbf{x}}) \cap \mathscr{I} \ , \\ \nabla_{\mathbf{x}} g_i(\bar{\mathbf{x}})\mathbf{v} &= 0,\ i \in \mathscr{A}(\bar{\mathbf{x}}) \cap \mathscr{E} \ . \end{aligned}$$

Constraint qualifications are sufficient conditions that ensure that the linearized feasible direction set at some point $\mathbf{x}$, $\mathscr{F}(\mathbf{x})$, is a good approximation of the feasible set $\mathscr{X}_F$ [73]. It naturally follows that when the constraint functions are linear, i.e. of the form $\mathbf{x} \mapsto g_i(\mathbf{x}) = \mathrm{b}_i^\intercal \mathbf{x} + c_i$, $\mathrm{b}_i \in \mathbb{R}^n$, $c_i \in \mathbb{R}$, no constraint qualification is required.

Optimality conditions are usually formulated on the *Lagrangian* (or *Lagrange function*) which is defined as follows

$$\mathscr{L}(\mathbf{x}, \gamma) = f(\mathbf{x}) + \sum_{i \in \mathscr{E} \cup \mathscr{I}} [\gamma]_i g_i(\mathbf{x}) \ , \tag{4.4}$$

[1] We consider that gradients are row vectors.

where $\mathbf{x} \in \mathscr{X}$ and γ is the vector of *Lagrange factors* $[\gamma]_i \in \mathbb{R}$, $i \in \mathscr{E} \cup \mathscr{I}$.

Remark 2. The vector of Lagrange factors is usually denoted λ in the literature. However, to avoid confusion with the population size parameter λ of evolutionary algorithms (see Chapter 3), we denote it γ here.

The following theorem states first-order necessary conditions of optimality. These conditions are often referred to as *Karush-Kuhn-Tucker* (KKT) conditions.

Theorem 3 (First-Order Necessary Conditions)**.** *Assume that $\boldsymbol{x}^*$ is a local optimum of the optimization problem in (4.1), that the functions f and g_i in (4.1) are continuously differentiable, and that some constraint qualification is satisfied at $\boldsymbol{x}^*$. Then, there exists a vector γ^* of Lagrange multipliers $[\gamma^*]_i$, $i \in \mathscr{E} \cup \mathscr{I}$, such that $[\boldsymbol{x}^*, \gamma^*]$ satisfies the following conditions:*

$$\nabla_{\boldsymbol{x}} \mathscr{L}(\boldsymbol{x}^*, \gamma^*) = \mathbf{0}, \tag{4.5a}$$
$$g_i(\boldsymbol{x}^*) = 0, \; i \in \mathscr{E}, \tag{4.5b}$$
$$g_i(\boldsymbol{x}^*) \leq 0, \; i \in \mathscr{I}, \tag{4.5c}$$
$$\gamma_i^* \geq 0, \; i \in \mathscr{I}, \tag{4.5d}$$
$$\gamma_i^* g_i(\boldsymbol{x}^*) = 0, \; i \in \mathscr{E} \cup \mathscr{I} \quad . \tag{4.5e}$$

The first condition (4.5a) can be written as $\nabla_{\mathbf{x}} f(\mathbf{x}^*) = -\sum_{i \in \mathscr{E} \cup \mathscr{I}} [\gamma^*]_i \nabla_{\mathbf{x}} g_i(\mathbf{x}^*)$. It states that the gradient of the objective function at a local optimum, $\nabla_{\mathbf{x}} f(\mathbf{x}^*)$, must be parallel to a linear combination of the constraint normals at $\mathbf{x}^*$, $\nabla_{\mathbf{x}} g_i(\mathbf{x}^*)$. Conditions (4.5e), called *complementary conditions*, imply either that a constraint is active or that the corresponding Lagrange multiplier $[\gamma^*]_i = 0$. Consequently, Lagrange multipliers corresponding to inactive constraints are always zero. Theorem 3 ensures the existence of a vector of Lagrange multipliers γ^* at a local optimum $\mathbf{x}^*$. If the LICQ holds, γ^* is unique [73]. It is also worth mentioning that if the objective function f is convex quadratic and the constraint functions are linear, the KKT necessary conditions are *sufficient* conditions for optimality [73, Theorem 16.4].

Before giving second-order optimality conditions, we introduce the notion of *critical cone* [73].

Definition 13 (Critical Cone)**.** Let $\mathbf{x}^*$ be a local optimum of the problem in (4.1) and let the KKT conditions be satisfied for some vector γ^* of Lagrange multipliers. Given the set $\mathscr{F}(\mathbf{x}^*)$

of linearized feasible directions at $\mathbf{x}^*$, the *critical cone* $\mathscr{C}(\mathbf{x}^*)$ is defined as

$$\mathscr{C}(\mathbf{x}^*) = \{\mathbf{d} \in \mathscr{F}(\mathbf{x}^*) \mid \nabla_{\mathbf{x}} g_i(\mathbf{x}^*)\mathbf{d} = 0, \text{ for all } i \in \mathscr{A}(\mathbf{x}^*) \cap \mathscr{I} \text{ such that } [\gamma^*]_i > 0\} \ ,$$

or equivalently as

$$\mathscr{C}(\mathbf{x}^*) = \left\{ \mathbf{d} \mid \begin{array}{ll} \nabla_{\mathbf{x}} g_i(\mathbf{x}^*)\mathbf{d} = 0, & \text{for all } i \in \mathscr{E}, \\ \nabla_{\mathbf{x}} g_i(\mathbf{x}^*)\mathbf{d} = 0, & \text{for all } i \in \mathscr{A}(\mathbf{x}^*) \cap \mathscr{I} \text{ such that } [\gamma^*]_i > 0, \\ \nabla_{\mathbf{x}} g_i(\mathbf{x}^*)\mathbf{d} \leq 0, & \text{for all } i \in \mathscr{A}(\mathbf{x}^*) \cap \mathscr{I} \text{ such that } [\gamma^*]_i = 0. \end{array} \right\} \ .$$

From Definition 13 and the first KKT condition (4.5a), we have

$$\mathbf{d} \in \mathscr{C}(\mathbf{x}^*) \Rightarrow \sum_{i \in \mathscr{E} \cup \mathscr{I}} [\gamma^*]_i \nabla_{\mathbf{x}} g_i(\mathbf{x}^*)\mathbf{d} = \nabla_{\mathbf{x}} f(\mathbf{x}^*)\mathbf{d} = 0 \ .$$

Therefore, the critical cone $\mathscr{C}(\mathbf{x}^*)$ contains those directions in $\mathscr{F}(\mathbf{x}^*)$ for which one cannot decide whether the first-order approximation of the objective function f will increase or decrease. The following theorem gives second-order necessary conditions of optimality.

Theorem 4 (Second-Order Necessary Conditions)**.** *Assume that $\boldsymbol{x}^*$ is a local optimum of the optimization problem in (4.1), that the functions f and g_i are twice continuously differentiable, and that the LICQ holds at $\boldsymbol{x}^*$. Let γ^* be a Lagrange vector satisfying the KKT conditions in (4.5). Then*

$$\boldsymbol{d}^\intercal \nabla^2_{\boldsymbol{xx}} \mathscr{L}(\boldsymbol{x}^*, \gamma^*)\boldsymbol{d} \geq 0, \ \textit{for all } \boldsymbol{d} \in \mathscr{C}(\boldsymbol{x}^*) \ .$$

The following theorem gives a second-order *sufficient* condition which—if satisfied—ensures that a given feasible point is a strict local optimum of (4.1).

Theorem 5 (Second-Order Sufficient Conditions)**.** *Assume that the functions f and g_i are twice continuously differentiable and that the KKT conditions in (4.5) are satisfied for some point $\boldsymbol{x}^*$ and for some vector γ^* of Lagrange multipliers. If*

$$\boldsymbol{d}^\intercal \nabla^2_{\boldsymbol{xx}} \mathscr{L}(\boldsymbol{x}^*, \gamma^*)\boldsymbol{d} > 0, \ \textit{for all } \boldsymbol{d} \in \mathscr{C}(\boldsymbol{x}^*), \ \boldsymbol{d} \neq \mathbf{0} \ ,$$

then $\boldsymbol{x}^*$ is a strict local optimum for (4.1).

Notice that no constraint qualification is required in Theorem 5. Also, the inequality is strict compared to Theorem 4.

Remark 3. All the necessary and sufficient conditions of optimality presented above hold if the objective function f and the constraint functions g_i are continuously differentiable in just an open set that contains the local optimum $\mathbf{x}^*$ [18]. Optimality conditions can also be expressed in the case of non-smooth functions which are continuous but non-differentiable everywhere. In such cases, the notion of gradient is replaced by those of *subgradient* or *generalized gradient* [44].

Duality

Duality is an important principle in mathematical optimization. Given an optimization problem, referred to as the *primal problem* (or simply *primal*), duality theory aims at (i) constructing an alternative optimization problem, the *dual problem* (or *dual*), from the primal problem, as well as (ii) relating the solutions of the primal and dual problems. In nonlinear programming, duality has given birth to some important algorithms as augmented Lagrangian approaches which we present later in this chapter. Most results of duality theory are expressed for the specific case of a convex objective function f and convex constraint functions g_i. Restricting ourselves to an optimization problem with only inequality constraints, we briefly present some of the main results in duality theory.

Given the optimization problem (4.1) where we only consider inequality constraints, a dual problem is given by

$$\begin{aligned} &\max_{\gamma} \; q(\gamma) \\ &\text{subject to } [\gamma]_i \geq 0, \; i \in \mathscr{I} \quad , \end{aligned} \tag{4.6}$$

where $q : \gamma \mapsto \min_{\mathbf{x}} \mathscr{L}(\mathbf{x}, \gamma)$ and $\mathscr{L}$ is the Lagrangian corresponding to our primal problem. It can easily be shown that the function q is concave [73]. Additionally, the dual (4.6) gives a lower bound on the optimal f-value of the primal, as stated in the following theorem.

Theorem 6 (Weak Duality)**.** *For any feasible point $\bar{\mathbf{x}}$ and any vector $\bar{\gamma}$ such that $[\bar{\gamma}]_i \geq 0$, $i \in \mathscr{I}$, we have $q(\bar{\gamma}) \leq f(\bar{\mathbf{x}})$.*

The next two results strongly rely on the convexity of the objective function and the constraint functions to define the relations between solutions of the primal and the dual.

Theorem 7. *Let $\bar{\mathbf{x}}$ be an optimum of the primal problem in (4.1) with only inequality constraints and assume that f and g_i, $i \in \mathscr{I}$, are convex and differentiable at $\bar{\mathbf{x}}$. Then, any vector $\bar{\gamma}$ such that $[\bar{\mathbf{x}}, \bar{\gamma}]$ satisfies the KKT conditions in (4.5) (with $\mathscr{E} = \emptyset$) is an optimum of the dual problem in (4.6).*

The following theorem shows how the optimum of the dual problem can be used to deduce the optimum of the primal problem [73].

Theorem 8. *Assume that f and g_i, $i \in \mathscr{I}$, are convex and continuously differentiable and that $\bar{\boldsymbol{x}}$ is an optimum of the primal problem (4.1) (where we consider $\mathscr{E} = \emptyset$) at which the LICQ holds. If $\hat{\boldsymbol{\gamma}}$ is an optimum of the dual problem (4.6) such that the minimum of $\mathscr{L}(\boldsymbol{x}, \hat{\boldsymbol{\gamma}})$ is attained for some $\hat{\boldsymbol{x}}$, then $\bar{\boldsymbol{x}} = \hat{\boldsymbol{x}}$ and $f(\bar{\boldsymbol{x}}) = \mathscr{L}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\gamma}})$.*

4.2 Constraint Handling Methods

Depending on the community, different classifications of constraint handling methods exist. In the first part of this section, we present four well-known constraint handling approaches in the community of mathematical nonlinear programming. Although powerful methods have been developed for the specific cases of linear and quadratic programming, we focus on nonlinear programming methods since they are more compatible with the black-box scenario. In the second part, we briefly review some constraint handling approaches for evolutionary algorithms.

4.2.1 Methods for Nonlinear Programming

We present in the sequel four families of constraint handling algorithms for nonlinear constrained optimization, namely *penalty function methods*, *augmented Lagrangian methods*—which can be seen as a particular case of the former family, *sequential quadratic programming*, and *interior-point methods*.

Penalty Methods

Penalty methods [73, 44, 18, 84] date back to the 1950s. They transform a constrained optimization problem into a sequence of unconstrained optimization problems by constructing a new objective function, the *penalty function*, as the sum of the original objective function and positive terms[2] corresponding to constraint violations, weighted by a positive *penalty factor*. An example of penalty functions is the *quadratic penalty function* used for optimization problems with only equality constraints, and defined as

$$p(\mathbf{x}, \omega) = f(\mathbf{x}) + \frac{\omega}{2} \sum_{i \in \mathscr{E}} g_i(\mathbf{x})^2 \quad , \tag{4.7}$$

[2] We consider a constrained minimization problem.

for objective function f and equality constraints $g_i(\mathbf{x}) = 0$, $i \in \mathscr{E}$. The parameter $\omega > 0$ is the penalty factor and the penalization is quadratic and corresponds to $g_i(\mathbf{x})^2$ for each constraint. That way, each point $\mathbf{x}$ such that $g_i(\mathbf{x}) \neq 0$, for some $i \in \mathscr{E}$, is penalized. Note that a different penalty factor can be used for each constraint function. Convergence of the quadratic penalty method to the global optimum of the constrained problem is proven under the assumption that the penalty parameter ω goes to infinity [44, 73]; therefore, ω is typically increased at each iteration in practice. Increasing the penalty factor, however, results in an ill-conditioned—therefore difficult—unconstrained optimization problem. A recent review of update rules for the penalty parameter is provided in [17]. Other penalty methods aim at constructing an *exact* penalty function, that is, for certain values of the penalty factor, the optimum of the constructed penalty function corresponds to the optimum of the constrained optimization problem [73, 47, 43, 40]. Such methods often use the derivatives of the objective function and the constraint functions.

Augmented Lagrangian Methods

Augmented Lagrangian methods [73, 44] are at the heart of this work. They were first introduced in [56, 76] as an alternative to penalty functions which suffer from ill-conditioning. Similarly to penalty function methods, augmented Lagrangian methods transform a constrained optimization problem into a sequence of unconstrained optimization problems. For the sake of illustration, let us consider the optimization problem in (4.1) where we consider only equality constraints, and the quadratic penalty function in (4.7). The idea of augmented Lagrangian methods is to reformulate the penalty function p by introducing a new parameter γ that emulates the vector of Lagrange multipliers, according to

$$h(\mathbf{x}, \gamma, \omega) = f(\mathbf{x}) + \sum_{i \in \mathscr{E}} [\gamma]_i g_i(\mathbf{x}) + \frac{\omega}{2} \sum_{i \in \mathscr{E}} g_i(\mathbf{x})^2 \quad , \tag{4.8}$$

where h is called the *augmented Lagrangian* and is a combination of the Lagrangian in (4.4) (considering that $\mathscr{I} = \emptyset$) and quadratic penalty terms. The parameter γ is the vector of *Lagrange factors* $[\gamma]_i$, $i \in \mathscr{E}$. Under differentiability assumptions, we have

$$\nabla_{\mathbf{x}} h(\mathbf{x}, \gamma, \omega) = \nabla_{\mathbf{x}} f(\mathbf{x}) + \sum_{i \in \mathscr{E}} ([\gamma]_i + \omega g_i(\mathbf{x})) \nabla_{\mathbf{x}} g_i(\mathbf{x}) \quad . \tag{4.9}$$

It is easy to see from (4.9) that for a point $\mathbf{x}^*$ satisfying the KKT conditions and for the corresponding vector γ^* of Lagrange multipliers, $\nabla_{\mathbf{x}} h(\mathbf{x}^*, \gamma^*, \omega) = \mathbf{0}$, for all $\omega > 0$; that is, $\mathbf{x}^*$ is a stationary point for $h(\mathbf{x}, \gamma^*, \omega)$. Let us now consider an algorithm that iteratively minimizes the augmented Lagrangian with respect to $\mathbf{x}$ for fixed values of γ and ω. Assuming

$\mathbf{x}_t$ is an approximate minimum for (4.8) at iteration t, the following holds [73]:

$$\nabla_{\mathbf{x}} h(\mathbf{x}_t, \gamma_t, \omega_t) = \nabla_{\mathbf{x}} f(\mathbf{x}_t) + \sum_{i \in \mathscr{E}} ([\gamma_t]_i + \omega_t g_i(\mathbf{x}_t)) \nabla_{\mathbf{x}} g_i(\mathbf{x}_t) \approx \mathbf{0} \ .$$

Therefore, if $\mathbf{x}_t$ satisfies the KKT conditions and if γ^* is the corresponding vector of Lagrange multipliers, then

$$[\gamma^*]_i \approx [\gamma_t]_i + \omega_t g_i(\mathbf{x}_t) \Leftrightarrow g_i(\mathbf{x}_t) \approx \frac{1}{\omega_t} ([\gamma^*]_i - [\gamma_t]_i) \ ,$$

for all $i \in \mathscr{E}$. This relation intuitively suggests that if $[\gamma_t]_i \to [\gamma^*]_i$, for all $i \in \mathscr{E}$, then $g_i(\mathbf{x}_t) \to 0$, without the need for ω_t to go to infinity [44, 73].

We consider in this work *adaptive* augmented Lagrangian approaches where the Lagrange factors are updated to converge to actual Lagrange multipliers and where penalty factors are updated to guide the search towards feasible solutions, ideally without unnecessarily increasing the conditioning of the problem at hand. The classical update rule for γ used with the augmented Lagrangian in (4.8) is given by [73, 44]

$$[\gamma_{t+1}]_i = [\gamma_t]_i + \omega_t g_i(\mathbf{x}_{t+1}), \text{ for all } i \in \mathscr{E} \ .$$

A broader discussion on augmented Lagrangian formulations and update rules for Lagrange factors is given in [73].

Augmented Lagrangian methods have drawn increasing attention since their introduction in the 1960s: in [32, 65, 22], the convergence of different augmented-Lagrangian-based algorithms for nonlinear programming is investigated and in [23], augmented Lagrangian methods for practical optimization problems are discussed.

Sequential Quadratic Programming

Sequential quadratic programming (SQP) [24, 73] is an approach for solving nonlinear constrained optimization problems that consists in iteratively solving quadratic approximations of the original problem. The solution of the current quadratic model is used to construct a more accurate model at the next iteration. The performance of a SQP method relies on the accuracy of the quadratic approximations as well as on the efficiency of the algorithm used to solve the quadratic subproblems. SQP can be combined with other methods for nonlinear programming, such as interior-point methods and active set methods presented below.

Interior-Point Methods

Modern interior-point (or barrier) methods [42, 75, 73] date back to 1984 when a polynomial time interior-point algorithm for linear programming was presented in [59]. They have been since largely studied in theory and extended to nonlinear programming. The idea of interior-point methods is to generate solutions that satisfy *strict* inequality constraints. More precisely, given the constrained optimization problem in (4.1), inequality constraints are transformed into equality constraints by introducing a vector s of slack variables as follows [73]:

$$\begin{aligned} g_i(\mathbf{x}) - [s]_i &= 0, \ i \in \mathscr{I} \\ [s]_i &\geq 0, \ i \in \mathscr{I} \quad . \end{aligned}$$

The objective function $\min_{\mathbf{x},s} f(\mathbf{x}) - \mu \sum_{i \in \mathscr{I}} \ln[s]_i$ is considered instead of f and the original constrained problem (4.1) is replaced with

$$\begin{aligned} \min_{\mathbf{x},s} \ & f(\mathbf{x}) - \sum_{i \in \mathscr{I}} \ln[s]_i \\ \text{subject to } & g_i(\mathbf{x}) = 0, \ i \in \mathscr{E} \\ & g_i(\mathbf{x}) - [s]_i = 0, \ i \in \mathscr{I} \quad , \end{aligned} \tag{4.10}$$

where $\mu > 0$ and where the term $-\sum_{i \in \mathscr{I}} \ln[s]_i$ prevents $[s]_i$ from getting too close to 0. Interior-point methods iteratively compute steps by solving the system of equations given by the KKT conditions for (4.10) (see (4.5)) with Newton's method. To avoid computing derivatives, some interior-point methods construct a quadratic model of (4.10).

Active Set Methods

Active set methods [44, 73] were first described for quadratic programming, as an extension of Dantzig's simplex method [73] for linear programming. They try to estimate the active set $\mathscr{A}(\mathbf{x}^*)$ (see Definition 9) at an optimum $\mathbf{x}^*$ of the constrained problem in (4.1); the idea is that if $\mathscr{A}(\mathbf{x}^*)$ is known beforehand, one can simply consider the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \ & f(\mathbf{x}) \\ \text{subject to } & g_i(\mathbf{x}) = 0, \ i \in \mathscr{A}(\mathbf{x}^*) \quad . \end{aligned}$$

Given the current estimate of $\mathbf{x}^*$ at iteration t, $\mathbf{x}_t$, and the corresponding active set $\mathscr{A}(\mathbf{x}_t)$, the algorithm tries to find a better feasible solution for the optimization problem

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ &\text{subject to } g_i(\mathbf{x}) = 0,\ i \in \mathscr{A}(\mathbf{x}_t) \quad , \end{aligned} \tag{4.11}$$

where active inequality constraints are regarded as equality constraints and where the remaining non-active constraints are ignored. If no better solution is found and $\mathbf{x}_t$ satisfies the KKT conditions corresponding to (4.11), the algorithm stops and returns $\mathbf{x}_t$. If negative Lagrange multipliers are computed, active inequality constraints corresponding to them are removed from the active set. If a better solution than $\mathbf{x}_t$ is found, the active set is updated by adding active constraints at the new solution and the process is repeated [73]. A popular class of active set methods, the so-called *active set sequential quadratic programming* methods [73], compute the Lagrange multipliers of a quadratic model of (4.11).

4.2.2 Constraint Handling In Evolutionary Algorithms

An overview of most recent constraint handling techniques for evolutionary algorithms is provided in [69, 30]. Following the classification in [30], a constraint handling method falls into one of the following four categories:

Penalty Functions

Penalty functions (or methods) are described in Subsection 4.2.1. The simplest penalty function is the so-called *death penalty*, or *resampling*, where unfeasible solutions are rejected and generated again. In [2, 27, 28], the behavior of different $(1,\lambda)$-ESs with resampling is investigated on the linear function subject to one linear constraint. Despite its simplicity, resampling can be very costly in practice. Moreover, it cannot handle equality constraints properly. We also distinguish between *static penalty functions* which keep the penalty parameter fixed, *dynamic penalty functions* which update the penalty parameter at every iteration, and *adaptive penalty functions* which update the penalty parameter using information from sampled candidate solutions. In [31], the authors present an adaptive-penalty-based constraint handling mechanism for CMA-ES [53] for the problem of designing a space launcher. A multiplicative update is used to adapt the penalty factors, where a factor is increased if the ratio of feasible solutions in the corresponding constraint is smaller than a target probability and decreased otherwise. Results indicate an increasing ratio of feasible solutions as the optimization progresses.

Augmented Lagrangian methods (see Subsection 4.2.1) are particular penalty functions. In [87], the authors present a coevolutionary algorithm for solving the dual problem expressed as a function of the augmented Lagrangian; to that end, two populations—one for the parameter vector and one for Lagrange factors—are evolved using an evolution strategy with self-adaptation. In [36], an augmented-Lagrangian-based genetic algorithm for constrained optimization is described. To converge to an optimal solution, a local search procedure is used to improve the current best solution. In [5], an adaptive augmented Lagrangian constraint handling approach is implemented for a $(1+1)$-ES for the case of one inequality constraint, and an adaptation rule for the penalty factor is presented. Numerical tests on the sphere function and on a moderately ill-conditioned ellipsoid function show linear convergence of the algorithm.

Special Representations and Operators

Methods that fall into this category try to preserve feasibility of a solution either by mapping unfeasible solutions into the feasible domain, or by mapping the entire feasible domain into a different space that is easier to explore. In [61], a homomorphous mapping between the feasible domain and a n-dimensional cube is presented. The implementation of special representations and operators, however, is often difficult in practice [69].

Separation of Constraints and Objectives

These methods handle the objective function and the constraints separately. For instance, some methods apply a multi-objective approach to solve a constrained problem by minimizing the objective function along with constraint violations [68].

Other Methods

The classification presented above is not exhaustive and other approaches exist in the literature. In [64], an evolutionary algorithm for solving the dual problem is presented as well as an exact penalty function. In [4], the authors design a constraint handling mechanism for CMA-ES: the idea is to estimate the normal vectors of the local constraints using unfeasible solutions, then to use this information to update the covariance matrix by reducing the variance in these directions. More recently, a $(1+1)$-ES with an active set constraint handling approach has been described in [3].

4.3 Discussion

An important part of this work is dedicated to the analysis of linear convergence of adaptive randomized algorithms for constrained optimization (see Chapter 6). Among the constraint handling methods presented in Section 4.2, we choose to investigate (adaptive) augmented Lagrangian methods. Besides their solid theoretical background [32, 65, 22], augmented Lagrangian methods optimize a sequence of unconstrained problems instead of the original constrained one. Therefore, it is possible to take advantage of the numerous efficient randomized algorithms for unconstrained optimization to solve a constrained problem. Another reason for this choice is the work of Arnold and Porter in [5] that features an augmented-Lagrangian-based $(1+1)$-ES that is observed to converge linearly on two simple convex quadratic functions with one linear inequality constraint. Their results encouraged us to investigate the observed linear convergence via Markov chain theory.

Chapter 5

Evaluating Step-Size Adaptation Mechanisms

In this chapter, we present our contributions related to the evaluation of step-size adaptation mechanisms in randomized algorithms for unconstrained optimization.

We recall that candidate solutions are sampled by an evolutionary algorithm as follows

$$\mathbf{X}_{t+1}^{i} = Sol((\mathbf{X}_t, \sigma_t), \mathbf{U}_{t+1}^{i}),\ i = 1, \cdots, \lambda \quad , \tag{5.1}$$

where the deterministic function $Sol : \mathbb{R}^n \times \mathbb{R}_{>}^{+} \times \mathbb{R}^n \to \mathbb{R}^n$ computes a candidate solution $\mathbf{X}_{t+1}^{i}$ using the current solution $\mathbf{X}_t$, the step-size σ_t, and the random vector $\mathbf{U}_{t+1}^{i}$. In practice, Sol is often defined as

$$Sol((\mathbf{x}, \sigma), \boldsymbol{u}) = \mathbf{x} + \sigma \boldsymbol{u} \quad . \tag{5.2}$$

It is then easy to see that the step-size σ_t controls the distance of a candidate solution from the current one, and therefore, the diversity within the population. Depending on the situation, an algorithm should maintain a large diversity (step-size) to efficiently explore the search space or quickly decrease the step-size in order to quickly converge to an optimum. Therefore, having a proper step-size adaptation mechanism is crucial for the performance of a practical algorithm.

A common practice in the literature is to assess step-size adaptation mechanisms on the sphere function, the motivation being that the sphere function is reasonably easy to study and that most functions can be approximated locally by a sphere. However, considering only the sphere function can be misleading as illustrated in Section 5.1 where we present a minimal methodology to assess a step-size adaptation algorithm more thoroughly. In Section 5.2, we present the results of benchmarking two step-size adaptation algorithms on the BBOB unconstrained testbed of the COCO test platform [52].

5.1 How to Assess Step-Size Adaptation Mechanisms in Randomised Search

In the following paper [51], we present a methodology for assessing step-size adaptation mechanisms in the case of unconstrained optimization. Our methodology consists in testing a given algorithm on a minimal set of functions, each representing a practical difficulty and, therefore, evaluating a particular feature of the algorithm. This work was published in the proceedings of the Parallel Problem Solving from Nature conference of 2014. The notations are slightly different from the ones adopted in the rest of this thesis (for instance, the current estimate of the optimum is denoted $\boldsymbol{x}^{(t)}$ in the paper instead of the usual $\mathbf{X}_t$).

How to Assess Step-Size Adaptation Mechanisms in Randomised Search

Nikolaus Hansen, Asma Atamna, and Anne Auger

Inria*
LRI (UMR 8623), University of Paris-Sud (UPSud), France

Abstract

Step-size adaptation for randomised search algorithms like evolution strategies is a crucial feature for their performance. The adaptation must, depending on the situation, sustain a large diversity or entertain fast convergence to the desired optimum. The assessment of step-size adaptation mechanisms is therefore non-trivial and often done in too restricted scenarios, possibly only on the sphere function. This paper introduces a (minimal) methodology combined with a practical procedure to conduct a more thorough assessment of the overall population diversity of a randomised search algorithm in different scenarios. We illustrate the methodology on evolution strategies with σ-self-adaptation, cumulative step-size adaptation and two-point adaptation. For the latter, we introduce a variant that abstains from *additional* samples by constructing two particular individuals within the *given* population to decide on the step-size change. We find that results on the sphere function alone can be rather misleading to assess mechanisms to control overall population diversity. The most striking flaws we observe for self-adaptation: on the linear function, the step-size increments are rather small, and on a moderately conditioned ellipsoid function, the adapted step-size is 20 times smaller than optimal.

1 Introduction

In this paper we consider a fitness or objective function, $f : \mathbb{R}^n \to \mathbb{R}$, to be minimised in a black-box optimisation scenario, and an evolutionary algorithm, or randomised search method, generating λ offspring according to

$$\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \boldsymbol{\sigma}^{(t)} \times \boldsymbol{z}_k^{(t)} \quad , \; k = 1, \ldots, \lambda \quad , \tag{1}$$

*Research centre Saclay–Île-de-France, TAO team, lastname@lri.fr

where $\boldsymbol{x}^{(t)} \in \mathbb{R}^n$ denotes the incumbent solution at iteration t and $\boldsymbol{z}_k^{(t)} \in \mathbb{R}^n$ are i.i.d. random vectors. The "overall variance" of the offspring population in (1) is determined by the diversity parameter $\sigma^{(t)}$. More generally, we rely on two assumptions: (i) we have a valid measurement for the "global diversity" of the offspring population, denoted as $\sigma^{(t)}$, and (ii) the shape of the offspring population (determined by the distribution of $\boldsymbol{z}_k^{(t)}$ in (1)) does not change remarkably during the investigated time range of t.

Controlling the overall diversity in the population plays a crucial role in randomised search and has been typically approached by step-size adaptation. Two conflicting objectives are in place. On the one hand, diversity should be as large as possible to prevent premature convergence or convergence to the very next local optimum. On the other hand, fast convergence to a global (or a good local) optimum is desired which is usually accompanied and facilitated by a fast decrease of diversity.

While adaptation of the *shape* of the sample distribution appears to be a solved problem in moderate dimension [6, 10, 11] (e.g. by CMA), the effective adaptation of the *overall* population diversity seems yet to pose open questions, in particular with recombination or without entire control over the realised distribution. For example, cumulative step-size adaptation is prone to fail when repair or rejection sampling is used.

In this context, we propose a basic *assessment procedure* to evaluate the capability of step-size control, or the entire search algorithm for that matter, to keep the overall diversity, or step-size $\sigma^{(t)}$, within reasonable limits. This procedure might be used during an algorithm designing process, however we like to remind the general scientific principle that a procedure used to systematically *tune parameters* of an algorithm is forfeited to *assess* the resulting algorithm.

In the next section we introduce the assessment methodology. Section 3 introduces the algorithms used in the case study in Section 4. We also introduce a simplified two-point adaptation and tune its damping parameter on the sphere function in Section 3. Section 5 provides a short discussion and summary.

2 Step-Size Evaluation Methodology

General demands on the behaviour of evolutionary algorithms were suggested previously, e.g. in [4, 11]. Here, we propose a methodology to specifically investigate and assess the overall population diversity, or step-size, towards meeting reasonable demands via the following scenarios:

Random fitness (and flat fitness). On the random fitness, all f-values, $f(\boldsymbol{x})$, are i.i.d., independently of $\boldsymbol{x}$ as a continuous random variable. For algorithms invariant under order-preserving transformations of f, i.e., algorithms based on *f-rankings* only (as those investigated in this paper), testing a single continuous f-distribution is sufficient. Generally, we desire stationarity or unbiasedness of parameters under random fitness [11] and here we expect to see an unbiased random walk in log-scale. For the flat fitness, where f is constant, we expect the same behaviour. In contrast, [4] argues for an exponentially increasing step-size on the flat fitness which, however, involves the risk of divergence when the selection

pressure is weak [7].

The linear function, where $f : \boldsymbol{x} \mapsto x_1$ is the prototypical instantiation (see paragraph *Invariance* below). A linear function tests whether and how quickly the diversity can increase. With step-size to zero, any smooth function appears to be an instantiation of the linear function (unless at a local optimum or saddle point) and the diversity should increase in this case. We demand a fast exponential increase, that is, a linear increase on the log-scale [4]. The rate should be at least comparable to the rate of decrease on the sphere function or at least a factor of 1.1 within n evaluations or at least a factor of 2 in n iterations.

The sphere function, $f : \boldsymbol{x} \mapsto \sum_{i=1}^{n} x_i^2 = \|\boldsymbol{x}\|^2$, is the most simple quadratic function, demanding a rapid decrease of the step-size. Arguably, no other function requires a faster step-size decrement. Step-size control should not reduce the fastest possible (optimal) convergence rate on the sphere function by more than a factor of about three.

To achieve linear (i.e. fast) convergence on the sphere function we need to have, at least approximately, $\sigma^{(t)} \propto f(\boldsymbol{x}^{(t)})^{1/2}$, implying that σ and $f^{1/2}$ converge at the same rate. More specifically, on the sphere function with isotropic sample distribution, there is a constant $\sigma^*_{\text{opt}}(n)$ such that the step-size

$$\sigma^{(t)} = \frac{\sigma^*_{\text{opt}}(n)}{n} \times f(\boldsymbol{x}^{(t)})^{1/2} \tag{2}$$

achieves optimal convergence speed and $\sigma^*_{\text{opt}}(n) = \Theta(n^0) = \Theta(1)$. When running a real algorithm, the proportionality can only be satisfied in a stochastic sense, i.e. the random variable $\sigma^{(t)}/f(\boldsymbol{x}^{(t)})^{1/2}$ is stable (for example when $\boldsymbol{x}^{(t)}/\sigma^{(t)}$ is an irreducible, recurrent and ergodic Markov Chain [3]).

A similar reasoning on $\sigma^{(t)}$ holds true on the ellipsoid function, where the direct link between $\sigma^{(t)}$ and $f(\boldsymbol{x}^{(t)})^{1/2}$ is less obvious, however presumed in the following to obtain the *optimal* convergence rates to compare with.

The ellipsoid function, $f : \boldsymbol{x} \mapsto \sum_{i=1}^{n} \alpha^{(i-1)/(n-1)} x_i^2$, is arguably the most basic function where, for $\alpha \neq 1$, an isotropic distribution of the new offspring is not optimal. The parameter α represents the condition number of the Hessian matrix of f.

With isotropic sample distribution in (1) and $\alpha > 10$, the realised convergence rates are roughly proportional to $10/\alpha$ [12]. Recalling that $f^{1/2}(\boldsymbol{x}^{(t)})$ and the optimal value for $\sigma^{(t)}$, are linked to each other (Eq. (2)), we observe that with larger α, when approaching the optimum, the optimal step-size changes more slowly (because the realised convergence rate is small). The task to *estimate the optimal step-size* becomes more relevant than the task to *follow the change* of the optimal step-size. In this paper, experiments are done for $\alpha = 1, 10, 100$.

The stationary sphere is an artificial model, resembling the sphere function in that an isotropic sample distribution is optimal, but with *stationary optimal step-size*. While the

sphere function tests the ability to decrease the step-size quickly, the stationary sphere function tests the ability to adapt the step-size close to the optimal step-size in the same sphere-like topography without approaching the optimum. With global intermediate or weighted recombination, as used below, the stationary sphere is simulated by setting the norm of the resulting recombination vector (super-parent) to one and re-normalisation of all other individuals or solutions in the algorithm's state by the same factor (see, e.g., lines 5–6 in Algorithm 3). When the population is never reduced to a single point, an appropriate normalisation factor needs to be identified (omitted due to space restrictions). The stationary sphere model is arguably the easiest model for step-size adaptation and we expect to observe close to the optimal step-sizes.

Convergence rate and optimal step-size. On the last three functions, we compute from a single run with t iterations the consistent estimator

$$\widehat{c} = -\frac{1}{T}\sum_{s=t-T}^{t-1}\frac{1}{2}\ln\left(\frac{f(\boldsymbol{x}^{(s+1)})}{f(\boldsymbol{x}^{(s)})}\right) \tag{3}$$

for the convergence rate [2, Eq. (24)], where $\boldsymbol{x}^{(s)} \in \mathbb{R}^n$ is the solution proposed at time step s, and the burn-in time $t-T$ diminishes the possible bias due to initialisation. In this paper we use $T = \lceil t/2 \rceil$, i.e. half of the overall time steps for aggregated measurements. If necessary (e.g., when we terminate due to numerical precision, but want more data), we average $\widehat{c}$ over several runs.

We obtain the values for the *optimal* step-size and convergence rate empirically by measuring the convergence rate with $\sigma^{(t)}$ set according to (2) and sweeping through different values for σ^*_{opt}. Generally, we demand the "real" algorithm to perform within a factor of three of this optimal convergence rate, and we prefer larger step-sizes to smaller ones, given the same performance is observed.

Invariance is an important concept in the assessment of algorithms. For example, all linear functions are identical for the below assessed algorithms, because the algorithms are invariant under affine transformations of f and under rotations of the search space. In the case where algorithms do not exhibit certain invariances (e.g. rotation invariance), it is advisable to test different instantiations (e.g. different rotations) of the above scenarios. Scale invariance on the other hand is a prerequisite to measure (3) independently of initial step-size or the distance to the optimum.

We now apply our methodology to three step-size adaptation methods. Due to the space limits, we do not always display single runs, but we consider investigating the evolution of f and σ (both displayed in the log scale) in single runs in all scenarios part of the assessment procedure [15].

3 Considered Step-Size Adaptation Methods

In the following, we consider the $(\mu/\mu,\lambda)$-ES with weighted recombination [11]. The offspring are generated as in (1) where the i.i.d. $\boldsymbol{z}_k^{(t)}$ follow the standard multivariate normal

Algorithm 1 The $(\mu/\mu,\lambda)$-σSA-ES

0 **given** $n \in \mathbb{N}_+$, λ, μ, w_k, $\tau = 1/\sqrt{2n}$
1 **initialize** $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, $\sigma^{(0)} \in \mathbb{R}_+$
2 **while** not happy
3 **if** *stationary_sphere* :
4 $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t)} / \|\boldsymbol{x}^{(t)}\|$
5 **for** $k \in \{1, \ldots, \lambda\}$
6 $\xi_k^{(t)} = \tau \mathcal{N}_{t,k}(0,1)$
7 $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$
8 $\sigma_k^{(t)} = \sigma^{(t)} \times \exp(\xi_k^{(t)})$
9 $\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \sigma_k^{(t)} \times \boldsymbol{z}_k^{(t)}$
10 $\sigma^{(t+1)} = \sum_{k=1}^{\mu} w_k \sigma_{k:\lambda}^{(t)}$
11 $\boldsymbol{x}^{(t+1)} = \sum_{k=1}^{\mu} w_k \boldsymbol{x}_{k:\lambda}^{(t)}$
12 $t = t + 1$

Algorithm 2 The $(\mu/\mu,\lambda)$-CSA-ES

0 **given** $n \in \mathbb{N}_+$, λ, μ, w_k, c_σ, d
1 **initialize** $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, $\sigma^{(0)} \in \mathbb{R}_+$, $\boldsymbol{p}_\sigma^{(0)} = \mathbf{0}$
2 **while** not happy
3 **if** *stationary_sphere* :
4 $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t)} / \|\boldsymbol{x}^{(t)}\|$
5 **for** $k \in \{1, \ldots, \lambda\}$
6 $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$
7 $\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \sigma^{(t)} \times \boldsymbol{z}_k^{(t)}$
8 $\boldsymbol{p}_\sigma^{(t+1)} = (1 - c_\sigma)\boldsymbol{p}_\sigma^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)/\sum_k^\mu w_k^2} \sum_{k=1}^{\mu} w_k \boldsymbol{z}_{k:\lambda}^{(t)}$
9 $\sigma^{(t+1)} = \sigma^{(t)} \times \exp^{\frac{c_\sigma}{d}} \left(\frac{\|\boldsymbol{p}_\sigma^{(t+1)}\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right)$
10 $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \sigma^{(t)} \sum_{k=1}^{\mu} w_k \boldsymbol{z}_{k:\lambda}^{(t)}$
11 $t = t + 1$

distributions, i.e., $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$. They are sorted according to their fitness such that

$$f\left(\boldsymbol{x}_{1:\lambda}^{(t)}\right) \leq f\left(\boldsymbol{x}_{2:\lambda}^{(t)}\right) \leq \cdots \leq f\left(\boldsymbol{x}_{\lambda:\lambda}^{(t)}\right) \quad , \tag{4}$$

thereby defining the index $k:\lambda$ used in the following. The μ best individuals are then recombined according to

$$\boldsymbol{x}^{(t+1)} = \sum_{k=1}^{\mu} w_k \boldsymbol{x}_{k:\lambda}^{(t)} \quad , \tag{5}$$

where w_k's are chosen to be optimal on the infinite-dimensional sphere function [1]. We set $\mu = \lfloor \lambda/2 \rfloor$ and therefore have only positive weights while $\lambda = 4 + \lfloor 3 \ln n \rfloor$.

We consider here three ways to adapt the step-size in (1). Self-Adaptation (SA) [14] and Cumulative Step-size Adaptation (CSA) [11] are given in Algorithm 1 and 2. The used default parameter settings for the latter are taken from [9] as $c_\sigma = \frac{\mu_w + 2}{n + \mu_w + 5}$, $d = 1 + 2\max\left(0, \sqrt{\frac{\mu_w - 1}{n+1}} - 1\right) + c_\sigma$. The third method considered for step-size adaptation is presented in the following.

Two-Point Step-Size Adaptation (TPA). We consider a tidied version of Two-Point Step-Size Adaptation (TPA) based on [8, 13]. Conceptually, TPA implements a very coarse line search along the direction of the latest mean shift from $\boldsymbol{x}^{(t-1)}$ to $\boldsymbol{x}^{(t)}$. In our version, we

Algorithm 3 The $(\mu/\mu,\lambda)$-ES with TPA

0 **given** $n \in \mathbb{N}_+$, λ, μ, $c_\sigma = 0.3$, $d_\sigma = \sqrt{n}$, w_k
1 **init** $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, $\sigma^{(0)} \in \mathbb{R}_+$, $t = 0$, $s^{(0)} = 0$
2 **while** not happy
3 **if** *stationary_sphere* :
4 **if** $t > 0$:
5 $\boldsymbol{x}^{(t-1)} = \boldsymbol{x}^{(t-1)} / \|\boldsymbol{x}^{(t)}\|$
6 $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t)} / \|\boldsymbol{x}^{(t)}\|$
7 **for** $k \in \{1, \ldots, \lambda\}$
8 $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$
9 **if** $t > 0$ and $k = 1$:
10 $\boldsymbol{z}_1^{(t)} = \|\mathcal{N}_t(\mathbf{0}, \mathbf{I})\| \times \dfrac{(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})}{\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\|}$
11 **if** $t > 0$ and $k = 2$:
12 $\boldsymbol{z}_2^{(t)} = -\boldsymbol{z}_1^{(t)}$
13 $\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \sigma^{(t)} \times \boldsymbol{z}_k^{(t)}$
14 $\boldsymbol{x}^{(t+1)} = \sum_{k=1}^{\mu} w_k \boldsymbol{x}_{k:\lambda}^{(t)}$
15 **if** $t > 0$:
16 $s^{(t)} = (1 - c_\sigma) s^{(t-1)} + c_\sigma \dfrac{\mathrm{rank}(\boldsymbol{x}_2^{(t)}) - \mathrm{rank}(\boldsymbol{x}_1^{(t)})}{\lambda - 1}$
17 $\sigma^{(t+1)} = \sigma^{(t)} \times \exp\left(\dfrac{s^{(t)}}{d_\sigma}\right)$
18 $t = t + 1$

sample the first two offspring *of the next iteration* along this line. These two offspring are generated as a mirrored pair, symmetric about the current mean vector $\boldsymbol{x}^{(t)}$,

$$\boldsymbol{x}_{1/2}^{(t)} = \boldsymbol{x}^{(t)} \pm \sigma^{(t)} \times \|\mathcal{N}_t(\mathbf{0}, \mathbf{I})\| \frac{\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}}{\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\|} \quad , \tag{6}$$

instead of (1). Their ranking according to the fitness is used to adapt the step-size: if $\boldsymbol{x}_1^{(t)}$ is better than $\boldsymbol{x}_2^{(t)}$ the step-size is increased, because there are better points in the direction of the mean shift vector, beyond of where the mean has been moved. Otherwise, the step-size is decreased. By using individuals that are likely to be sampled by the current distribution, information on the "signal strength" is available, because we can compare their fitness to the fitness of the remaining population. Accordingly, we take the difference between the f-ranks of $\boldsymbol{x}_1^{(t)}$ and $\boldsymbol{x}_2^{(t)}$ in the population, $\frac{\mathrm{rank}(\boldsymbol{x}_2^{(t)}) - \mathrm{rank}(\boldsymbol{x}_1^{(t)})}{\lambda - 1} \in [-1, 1]$. This normalised rank difference is averaged in $s^{(t)}$ and used to finally update the step-size $\sigma^{(t+1)} = \sigma^{(t)} \times \exp\left(s^{(t)}/d_\sigma\right)$, where the damping, d_σ, moderates the step-size changes. The details are shown in Algorithm 3.

The constant for which $\sigma^{(t)}$ in (2) achieves optimal convergence rate depends on the sampling. For TPA-like sampling, we denote it $\sigma^*_{\text{opt TPA}}$.

The Damping Factor. Here we identify a default value for the damping d_σ. To this aim, we follow a standard procedure: d_σ is tuned on the sphere function. For each value of d_σ, the algorithm is run 101 times with target f-value 10^{-8} (the f-value that stops the algorithm when reached), and if all runs reached the target within $10^5 n$ evaluations., the average number of f-evaluations is recorded, see Figure 1, left. We observe a steep incline to the left (small values of d_σ), where missing points indicate the failure of at least one run to

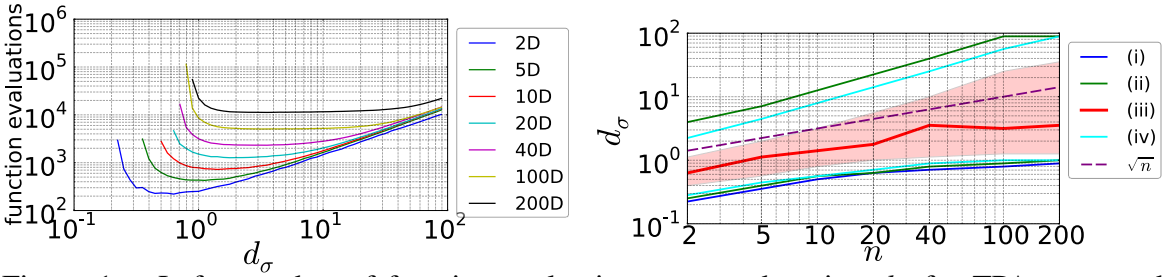

Figure 1: Left: number of function evaluations versus damping d_σ for TPA, averaged over 101 runs with target f-value 10^{-8}. Right: solid lines depict, from bottom to top, (i) the smallest damping where all runs reached the target value; (ii) the smallest and largest "reasonable" damping with a performance not worse than three times the best (lowest) value in the respective graph on the left; (iii) the damping with best performance, d_σ^*; (iv) the smallest and largest damping with performance no more than two times worse than the best value in the respective graph on the left, all plotted against dimension. The dashed line depicts $\sqrt{n}$. The filled area corresponds to damping values with at most 20% performance loss compared to the optimal damping.

reach the target after $10^5 n$ evaluations. To the right, the number of f-evaluations increases linearly with the damping and no failures are observed. We extract four damping values per dimension as shown and described in Figure 1, right. We then choose the damping to be (a) more than three times larger than the smallest "reasonable" value and (b) larger than the optimal value such that (c) reducing d_σ by a factor of two leads to a better performance than increasing it by a factor of two without (d) loosing more than a factor of two in performance compared to the best damping (see also [5]). The default choice becomes $d_\sigma = \sqrt{n}$. Note that we identified the damping only for the given default population size. The same procedure needs to be repeated to identify a damping parameter for different population sizes.

4 A Case Study

Experiments are conducted in dimensions between 2 and 100. The algorithms are run with the default parameter settings (Section 3) and initial $\boldsymbol{x}^{(0)} = (1, 0, \ldots, 0)^\mathsf{T}$. On random, linear, and ellipsoid function we have $\sigma^{(0)} = 1$, on the sphere and stationary sphere we have $\sigma^{(0)} = \sigma^*_{\text{opt}}/n$ (respectively $\sigma^*_{\text{opt TPA}}/n$) for SA and CSA (respectively TPA). Interquartile ranges are depicted as notched bars with the median at the notch.

Random Fitness. Figure 2 displays the evolution of $\sigma^{(t)}$ for 5000 iterations in 4- and 40-D, five runs for each algorithm. As expected by design, CSA and TPA show an unbiased random walk of $\log \sigma$, where TPA reveals a larger variance. In contrast, due to the combination of geometric mutation and arithmetic recombination of the step-sizes, the random walk of SA is biased [7] and $\log \sigma$ increases linearly with a rate of a little above (below) $10^{0.07} \approx 1.17$ in n iterations for $n = 40$ ($n = 4$, respectively).

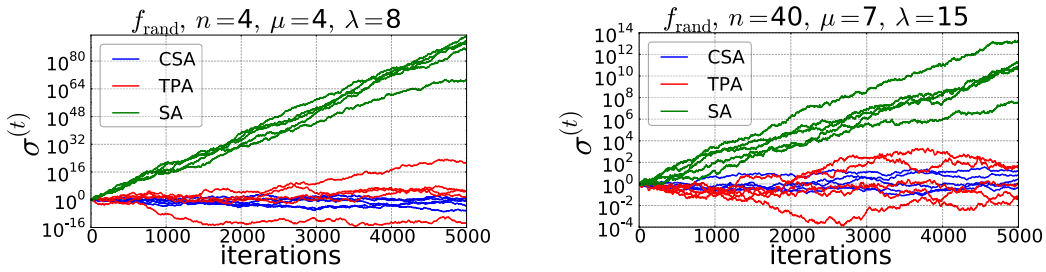

Figure 2: Evolution of $\sigma^{(t)}$ on the random fitness for 5 runs of SA (green), CSA (blue), and TPA (red) in 4-D (left) and 40-D (right).

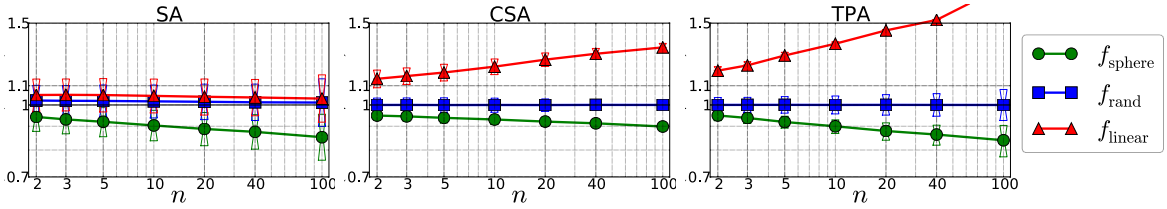

Figure 3: Step-size change after n evaluations, $(\sigma^{(t+1)}/\sigma^{(t)})^{n/\lambda}$, on the linear (red), the sphere (green), and the random function (blue).

Linear Function. On the linear function, the algorithms are run 100 times for 400 iterations. Figure 3 shows geometric average and quartiles of the step-size change realised after n evaluations, $(\sigma^{(t+1)}/\sigma^{(t)})^{n/\lambda}$, compared to results obtained on the random and the sphere function.

For CSA and TPA, the step-size increases by at least a factor of 1.14 within n evaluations. This factor increases slowly with increasing dimension (but never exceeds a factor of two) and the increment on the linear function is at least about three times faster than the decrement on the sphere function.

Self-Adaptation realises only an increment of a factor between 1.03 and 1.05 within n function evaluations, where also decrements appears frequently. The step-size grows faster than on the random function but up to four times slower than it shrinks on the sphere function. This latter observation, together with the observed slow changes rates, fails to meet our original demand.

Sphere. On the sphere function, the target f-value is 10^{-100}. Figure 4 shows nine single runs (left) with $\sigma^{(0)} = 10^{-5}$, the step-size as geometric average (middle), and the convergence rate $\widehat{c} \times n/\lambda$ (right, see (3)), both averaged over 100 runs.

All algorithms realise a too large step-size. In small dimensions, this leads to a loss in performance by about a factor of five, thereby failing our original demand. Fortunately, with increasing dimension the effect diminishes. For $n = 100$, TPA and SA reveal close to optimal convergence rates, whereas CSA is about two times slower.

Supposedly, we observe larger-than-optimal step-sizes, because the optimal step-size changes during the run and is therefore a *moving target*. Indeed, decreasing the damping parameters d or d_σ in CSA or TPA by a factor of two or increasing τ in SA improves the

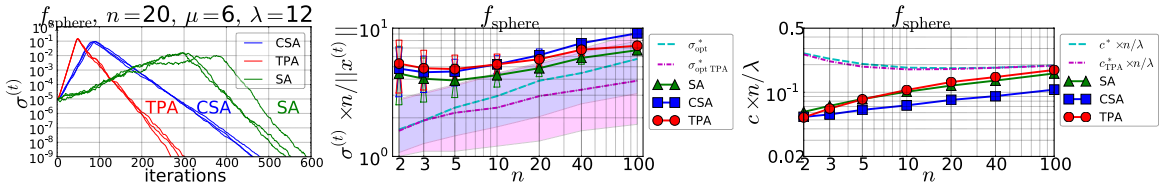

Figure 4: Single runs (left), step-size (middle) and convergence rate (right) on the sphere function, for SA (green), CSA (blue), and TPA (red) and the respective optimal values. Filled areas correspond to step-size values with at most 20% performance loss compared to σ_{opt} (or $\sigma_{\text{opt TPA}}$, respectively).

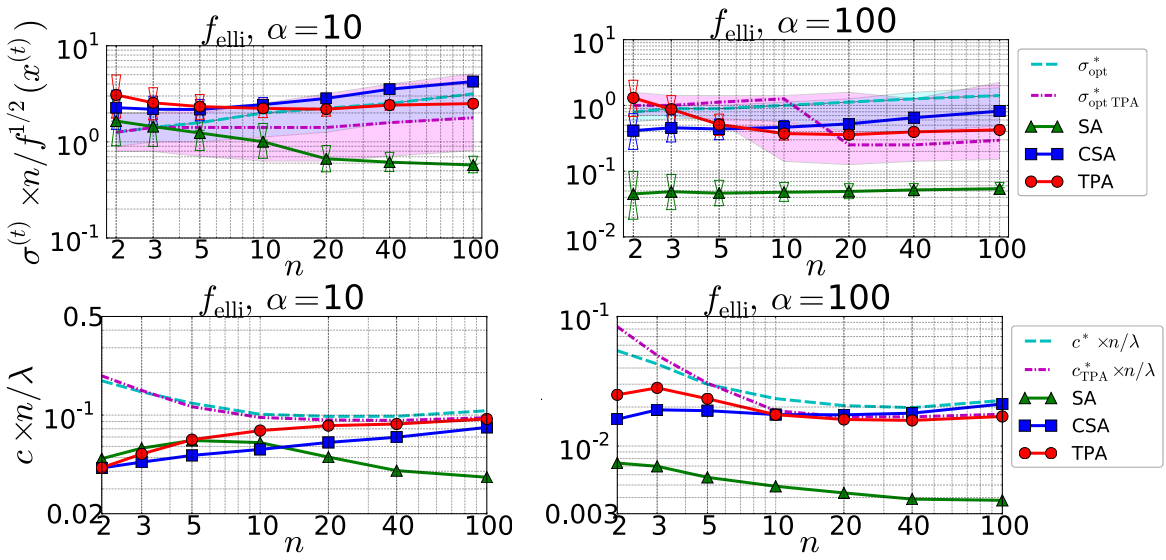

Figure 5: Results on the ellipsoid with condition number 10 (left) and 100 (right). Top: normalised step-size. Shaded areas depict the step-size range with at most 20% loss in convergence rate. Bottom: convergence rate according to (3).

convergence speeds thereby meeting just about the original demand. However for SA, this impairs the performance on the ellipsoid function with $\alpha = 10$.

Ellipsoid. Complementing the observations on the sphere function, which coincides with the ellipsoid function with $\alpha = 1$, the algorithms are investigated on the ellipsoid function with $\alpha \in \{10, 100\}$. These are very moderate condition numbers, where an isotropic distribution can still realise comparatively high convergence rates. We conduct 100 runs with target f-value[1] of 10^{-50}. Figure 5 shows the step-size as geometric average and the convergence rate $\widehat{c}$ from (3). With increasing condition number the realised step-sizes become across the board smaller (compared to the optimal step-size). For $\alpha = 10$, the step-size is still slightly too large with CSA and TPA, while SA shows already too small step-sizes. With $\alpha = 100$, SA realises a 20 times smaller than optimal step-size. Then, for $n \geq 10$, SA performs four to six times slower than optimal, while the other two methods reveal

[1] In general, we can use such a small target f-value only because the optimum is located at zero and because the distribution shape does not change over the iterations (see Section 1).

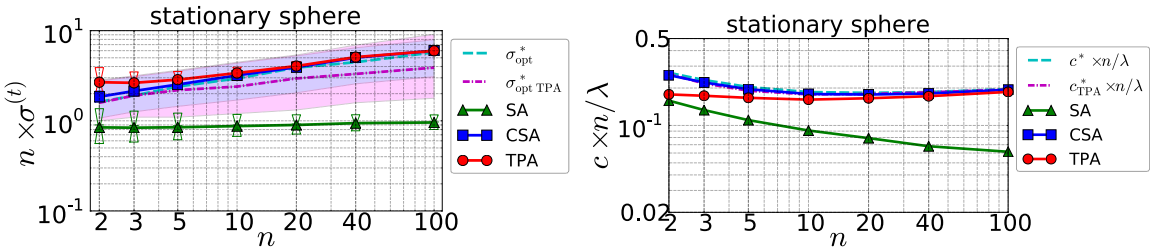

Figure 6: Step-size (left) and convergence rate (right) of SA (green), CSA (blue), and TPA (red) on the stationary sphere together with the respective optimal values. Shaded areas reflect step-sizes with no more than 20% loss in the achieved convergence rate.

close-to-optimal convergence rates.

Stationary Sphere. On the stationary sphere model, the algorithms are run for $t = 5000$ iterations. The convergence rate $\widehat{c}$ from (3) is estimated from 100 runs.

Figure 6 shows step-sizes (as geometric average) and convergence rates. The CSA achieves close to optimal step-sizes and convergence rates in all dimensions. The TPA reveals very similar step-sizes in larger dimensions, however for TPA they are somewhat too large, because the optimal step-size is somewhat smaller. Yet, only in smaller dimensions a (moderate) performance loss is observed.

In contrast, SA adapts always a too small step-size. The gap to the optimal step-size is a factor of two in 2-D and increases to a factor of 6 in 100-D. The loss in convergence rate is (slightly) above a factor of three only in 100-D. These observations are (qualitatively) similar to those on the ellipsoid function with condition number 100.

Compared to the sphere function, the observed step-sizes are *in all cases* considerably smaller, again supporting the hypothesis that too large step-sizes are observed on the sphere function mainly because the optimal step-size is a moving target.[2]

5 Discussion and Summary

We have introduced a methodology to assess the overall population diversity, for example determined via step-size adaptation, by describing the desired outcomes on basic scenarios. We conducted a case study assessing evolution strategies with weighted recombination and three different step-size adaptation mechanisms.

Despite the small number of investigated algorithms, we find in each test scenario, arguably with exception of the random function, limitations of at least one method: a (too) slow step-size increase on the linear function; a (too) slow step-size decrease on the sphere function in small dimensions; adaptation of a far too small step-size on the ellipsoid and stationary sphere. The results suggest that both, design and assessment of step-size adaptation methods is more intricate than one would have hoped for.

[2]Experiments with varying damping- or τ-values give additional strong support. Increasing damping impairs the performance on the sphere function (cp. Fig. 1) by reducing the change rate of the step-size, while it (slightly) improves the performance on the stationary sphere.

Acknowledgments.

This work was supported by the grant ANR-2012-MONU-0009 (NumBBO) of the French National Research Agency.

References

[1] D. V. Arnold. Optimal weighted recombination. In *Foundations of Genetic Algorithms*, pages 215–237. Springer, 2005.

[2] A. Auger and N. Hansen. Reconsidering the progress rate theory for evolution strategies in finite dimensions. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 445–452. ACM, 2006.

[3] A. Auger and N. Hansen. On Proving Linear Convergence of Comparison-based Step-size Adaptive Randomized Search on Scaling-Invariant Functions via Stability of Markov Chains, 2013. ArXiv eprint.

[4] H.-G. Beyer and K. Deb. On self-adaptive features in real-parameter evolutionary algorithms. *Evolutionary Computation, IEEE Transactions*, 5(3):250–270, 2001.

[5] D. Brockhoff, A. Auger, N. Hansen, D. V. Arnold, and T. Hohm. Mirrored sampling and sequential selection for evolution strategies. In *Parallel Problem Solving from Nature, PPSN XI*, pages 11–21. Springer, 2010.

[6] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber. Exponential natural evolution strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 393–400. ACM, 2010.

[7] N. Hansen. An analysis of mutative σ-self-adaptation on linear fitness functions. *Evolutionary Computation*, 14(3):255–275, 2006.

[8] N. Hansen. CMA-ES with two-point step-size adaptation. *CoRR*, abs/0805.0231, 2008.

[9] N. Hansen. The CMA evolution strategy: A tutorial. 2011.

[10] N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 282–291. Springer, 2004.

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

[12] 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.

[13] R. Salomon. Evolutionary algorithms and gradient search: Similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.

[14] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley Interscience, New York, 1995.

[15] http://hal.inria.fr/hal-00997294.

5.2 Benchmarking IPOP-CMA-ES-TPA and IPOP-CMA-ES-MSR on the BBOB Noiseless Testbed

With this paper [6], we illustrate how step-size adaptation can be assessed in practice. We benchmark for the first time two step-size adaptation algorithms on the BBOB noiseless testbed of the COCO platform [52]. The investigated algorithms are two-point step-size adaptation (TPA) [49] and median success rule (MSR) [1], paired with CMA-ES and the so-called IPOP (increasing population size) restart mechanism. This work was presented in the BBOB 2015 workshop. A former version of COCO was used for this work and the average runtime to reach a target value (aRT, see Subsection 3.5.1) was called the "expected running time" (ERT) at the time this work was conducted.

Benchmarking IPOP-CMA-ES-TPA and IPOP-CMA-ES-MSR on the BBOB Noiseless Testbed

Asma Atamna

Projet TAO, Inria
Saclay–Île-de-France
LRI, Bât. 660, Univ. Paris-Sud
91405 Orsay Cedex, France
atamna@lri.fr

Abstract

We benchmark IPOP-CMA-ES, a restart Covariance Matrix Adaptation Evolution Strategy with increasing population size, with two step-size adaptation mechanisms, Two-Point Step-Size Adapation (TPA) and Median Success Rule (MSR), on the BBOB noiseless testbed. We then compare IPOP-CMA-ES-TPA and IPOP-CMA-ES-MSR to IPOP-CMA-ES with the standard step-size adaptation mechanism, Cumulative Step-size Adaptation (CSA). We conduct experiments for a budget of 10^5 times the dimension of the search space. As expected, the algorithms perform alike on most functions. However, we observe some relevant differences, the most significant being on the attractive sector function where IPOP-CMA-TPA and IPOP-CMA-CSA outperform IPOP-CMA-MSR, and on the Rastrigin function where IPOP-CMA-MSR is the only algorithm to solve the function in all tested dimensions. We also observe that at least one of the three algorithms is comparable to the best BBOB-09 artificial algorithm on 13 functions.

1 Introduction

This paper compares three step-size adaptation methods coupled with IPOP-CMA-ES [2], a restarted version of the state-of-the-art Evolution Strategy (ES), the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [8], where the population size is increased for each restart, on the BBOB noiseless testbed [3, 7]. The step-size adaptation algorithms under consideration are Two-Point Step-Size Adaptation (TPA) [5], Median Success Rule (MSR) [1], and Cumulative Step-Size Adaptation (CSA) [8], the latter being the default

step-size adaptation method in CMA-ES. We first recall the general principle of the considered ES, we then describe the studied step-size adaptation algorithms, with a particular focus on TPA and MSR, and evaluate them empirically.

2 The $(\mu/\mu,\lambda)$-ES

In this paper, we consider the $(\mu/\mu,\lambda)$-ES with weighted recombination, where λ is the population size, μ is the number of parents, and ',' denotes non-elitist selection [4]. At iteration t, λ offspring, $\mathbf{X}_t^1,\ldots,\mathbf{X}_t^\lambda$, are sampled independently from a multivariate normal distribution according to

$$\mathbf{X}_t^i = \mathbf{X}_t + \sigma_t \mathcal{N}_t(0,\mathbf{C}_t) \quad ,i = 1,\ldots,\lambda \tag{1}$$

where $\mathcal{N}_t(0,\mathbf{C}_t)$ is the multivariate normal distribution with mean 0 and covariance matrix $\mathbf{C}_t$, σ_t is the step-size and defines the width of the sampling distribution. The μ best offspring are recombined to form the new solution

$$\mathbf{X}_{t+1} = \sum_{i=1}^{\mu} w_i \mathbf{X}_t^{i:\lambda} \tag{2}$$

where $\mathbf{X}_t^{i:\lambda}$ is the ith best offspring fitness-wise, $w_i > 0$ and $\sum_{i=1}^{\mu} w_i = 1$. In adaptive ES, σ_t and $\mathbf{C}_t$ are updated during the search process in order to achieve fast convergence.

3 IPOP-CMA-ES

IPOP-CMA-ES consists in launching independent restarts of CMA-ES by increasing the population size by a factor of two for each restart. Increasing the population size allows for a better covering of the search space and improves the performance of CMA-ES on multimodal functions [2]. The principle of the algorithm can be summed up in two steps:

1. run CMA-ES
2. if CMA-ES stops before reaching the target value and before exceeding the budget, double the population size and go to step 1

For a detailed description of the algorithm, see [2].

CMA-ES In this paper, we consider the $(\mu/\mu,\lambda)$-CMA-ES with weighted recombination, fully described in [8].

4 Step-Size Adaptation Methods

This section describes the three step-size adaptation methods under investigation.

4.1 TPA

In Two-Point Step-Size Adaptation, the first two offspring are sampled along the shift vector from the previous solution, $\mathbf{X}_{t-1}$, to the current solution $\mathbf{X}_t$, as a mirrored pair, symmetric to $\mathbf{X}_t$.

$$\mathbf{X}_t^{1,2} = \mathbf{X}_t \pm \sigma_t \times \|\mathcal{N}_t(0,\mathbf{I})\| \frac{\mathbf{X}_t - \mathbf{X}_{t-1}}{\|\mathbf{X}_t - \mathbf{X}_{t-1}\|} \tag{3}$$

where $\mathbf{I}$ is the identity matrix. We decide whether to increase or decrease the step-size σ_t depending on the fitness of $\mathbf{X}_t^1$ and $\mathbf{X}_t^2$: if $\mathbf{X}_t^1$ is better than $\mathbf{X}_t^2$, σ_t is increased as this indicates that there are better solutions in the direction of the latest solution shift. Otherwise, it is decreased. The following equations give the step-size update.

$$s_1 = (1 - c_\sigma)\, s_{t-1} + c_\sigma \frac{\mathrm{rank}(\mathbf{X}_t^2) - \mathrm{rank}(\mathbf{X}_t^1)}{\lambda - 1} \tag{4}$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{s_t}{d_\sigma}\right) \tag{5}$$

where $\mathrm{rank}(\mathbf{X}_t^i)$ is the fitness ranking of the ith individual among the entire population, $s_0 = 0$, $c_\sigma = 0.3$, and $d_\sigma = \sqrt{D}$ where D is the dimension of the search space. A more thorough description of the algorithm can be found in [5].

4.2 MSR

The Median Success Rule Step-Size Adaptation can be seen as a generalization of the $1/5$th success rule [10] to the case of $(\mu/\mu, \lambda)$-ES. The success is defined as the median individual (fitness-wise) of the current population, $\mathbf{X}_t^{m(\lambda)}$, being better than the jth best individual of the previous population, $\mathbf{X}_{t-1}^{j:\lambda}$. In practice, j is chosen such that the median success probability is approximately $1/2$ with optimal step-size on the sphere function [1]; this value corresponds to the 30th percentile. The idea is then to increase the step-size if $\mathbf{X}_t^{m(\lambda)}$ is fitter than $\mathbf{X}_{t-1}^{j:\lambda}$ and decrease it otherwise. The step-size σ_t is updated as

$$s_1 = (1 - c_\sigma)\, s_{t-1} + c_\sigma \frac{2}{\lambda}\left(K_{\mathrm{succ}} - \frac{\lambda}{2}\right) \tag{6}$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{s_t}{d_\sigma}\right) \tag{7}$$

where K_{succ} is the number of successful individuals, $s_0 = 0$, $c_\sigma = 0.3$, and $d_\sigma = 2 - 2/D$.

4.3 CSA

The Cumulative Step-Size Adaptation is the standard step-size adaptation method in CMA-ES. A detailed description of the method can be found in [8].

5 Experimental Procedure

We ran the algorithms with a budget of $10^5 \times D$ on the BBOB noiseless functions in six different dimensions. We used the python implementation of CMA-ES, cma 1.1.06. The source code can be found at [11]. TPA, MSR, and CSA are implemented in cma 1.1.06 as well as the IPOP restart strategy. For each run of the algorithms, the initial solution $\mathbf{X}_0$ is sampled uniformly in $[-4,4]^D$ and the initial step-size σ_0 is set to 2.5. The maximum number of restarts is set to 9. For all other parameters, default values are used (for instance, the population size $\lambda = 4 + \lfloor 3 \ln D \rfloor$ and the number of parents $\mu = \lambda/2$).

6 Results

Results from experiments according to [6] on the benchmark functions given in [3, 7] are presented in Figures 1, 3 and 4 and in Tables 1 and 2. The **expected running time (ERT)**, used in the figures and tables, depends on a given target function value, $f_t = f_{opt} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach f_t, summed over all trials and divided by the number of trials that actually reached f_t [6, 9]. **Statistical significance** is tested with the rank-sum test for a given target Δf_t using, for each trial, either the number of needed function evaluations to reach Δf_t (inverted and multiplied by -1), or, if the target was not reached, the best Δf-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

For the sake of simplicity, we will refer to IPOP-CMA-ES-TPA, IPOP-CMA-ES-MSR, and IPOP-CMA-ES-CSA as TPA, MSR, and CSA respectively in the following.

ERT versus dimension Figure 1 shows that in 5-D (respectively 20-D), TPA, MSR, and CSA solve 22 (respectively 19), 20 (respectively 20), and 22 (respectively 20) out of 24 functions. For unsolved functions (mainly multi-modal and weakly structured multi-modal functions), a larger budget is required (at least $10^6 \times D$ function evaluations). The algorithms have a comparable performance on most of the functions and scale similarly with the dimension. This corresponds to our expectations, as the three algorithms are very similar. On some functions, however, we observe relevant differences in the performance: on function 1 (sphere), TPA performs significantly better than MSR and CSA in at least one dimension. We also observe a significant difference on function 6 (attractive sector) where TPA and CSA outperform MSR in large dimensions. Single runs on function 6 show that MSR generates smaller step-sizes than TPA and CSA, which leads to its larger ERT. Figure 2 displays single runs of MSR (left) and CSA (right) in 20-D (due to space limitations, results for TPA are not presented). On function 3 (separable Rastrigin), MSR has the best performance. Our explanation is that having larger step-sizes than CSA and TPA on function 3 avoids getting stuck in local optima. On functions 16 (Weierstrass) and 19 (Griewank-Rosenbrock), TPA and CSA perform very similarly and better than MSR. On function 20 (Schwefel), CSA performs slightly better than TPA in small dimensions. The gap we see in 10-D between TPA and CSA is due to insufficient budget and should disappear by increasing the budget.

Another significant difference is observed on function 23 (Katsuuras) where MSR solves the function within the maximum budget and performs better than TPA and CSA. On function 21 (Gallagher 101 peaks), a larger budget is necessary to decide whether the observed difference is significant, since the ERTs are close to the maximum budget. Another observation is that each algorithm performs similarly on the original/rotated ellipsoid and Rosenbrock due to their rotational invariance. On Rastrigin functions, however, this is not the case, likely because the rotated function does not correspond to the original one.

Empirical cumulative distribution functions Figures 3 and 4 show the empirical cumulative distribution functions (ECDFs) of the number of function evaluations for 50 targets in dimensions 5 and 20 respectively. In 5-D, the ECDFs are quite similar for moderate and ill-conditioned functions. On separable functions, MSR solves about 82% of the problems for the fixed budget ($10^5 \times D$) while TPA and CSA solve about 73%. On multi-modal functions, TPA and CSA manage to solve all problems while MSR solves about 88% of the problems. While no algorithm solves all weakly structured multi-modal problems, TPA and CSA solve up to 76% of the problems for the maximum budget while MSR only solves about 56%. On the overall set of functions, TPA, MSR, and CSA solve roughly the same proportion of problems up to $10^4 \times D$ function evaluations. For the maximum budget, however, TPA and CSA solve about 90% of the problems while MSR only solves about 84%. In 20-D, two main differences are observed: firstly, TPA and CSA solve about 8% (respectively 10%) less separable (respectively multi-modal) problems than in 5-D (none of them managed to solve function 3 in 20-D). Secondly, CSA is better than MSR and TPA on weakly structured multi-modal problems and solves about 50% of the problems, being 10% more than MSR and 13% more than TPA.

7 Discussion

We evaluated IPOP-CMA-ES with two different and relatively new step-size adaptation schemes, TPA and MSR, on the BBOB noiseless continuous functions. We then compared them to IPOP-CMA-ES with the standard step-size adaptation method, CSA. As expected, empirical results showed that the three algorithms need nearly the same number of function evaluations in average to solve the target $f_t = f_{opt} + 10^{-8}$ on a large number of functions. However, significant differences were observed, the most notable were on the attractive sector function where TPA and CSA outperformed MSR in large dimensions and on Rastrigin where MSR was the best. 16 functions out of 24 were solved by all the algorithms in all dimensions while some multi-modal and weakly structured multi-modal functions remained unsolved because the chosen budget ($10^5 \times D$ function evaluations) was insufficient. On the other hand, the performance was comparable to the best BBOB-09 results on 13 functions for at least one algorithm, generally in large dimensions.

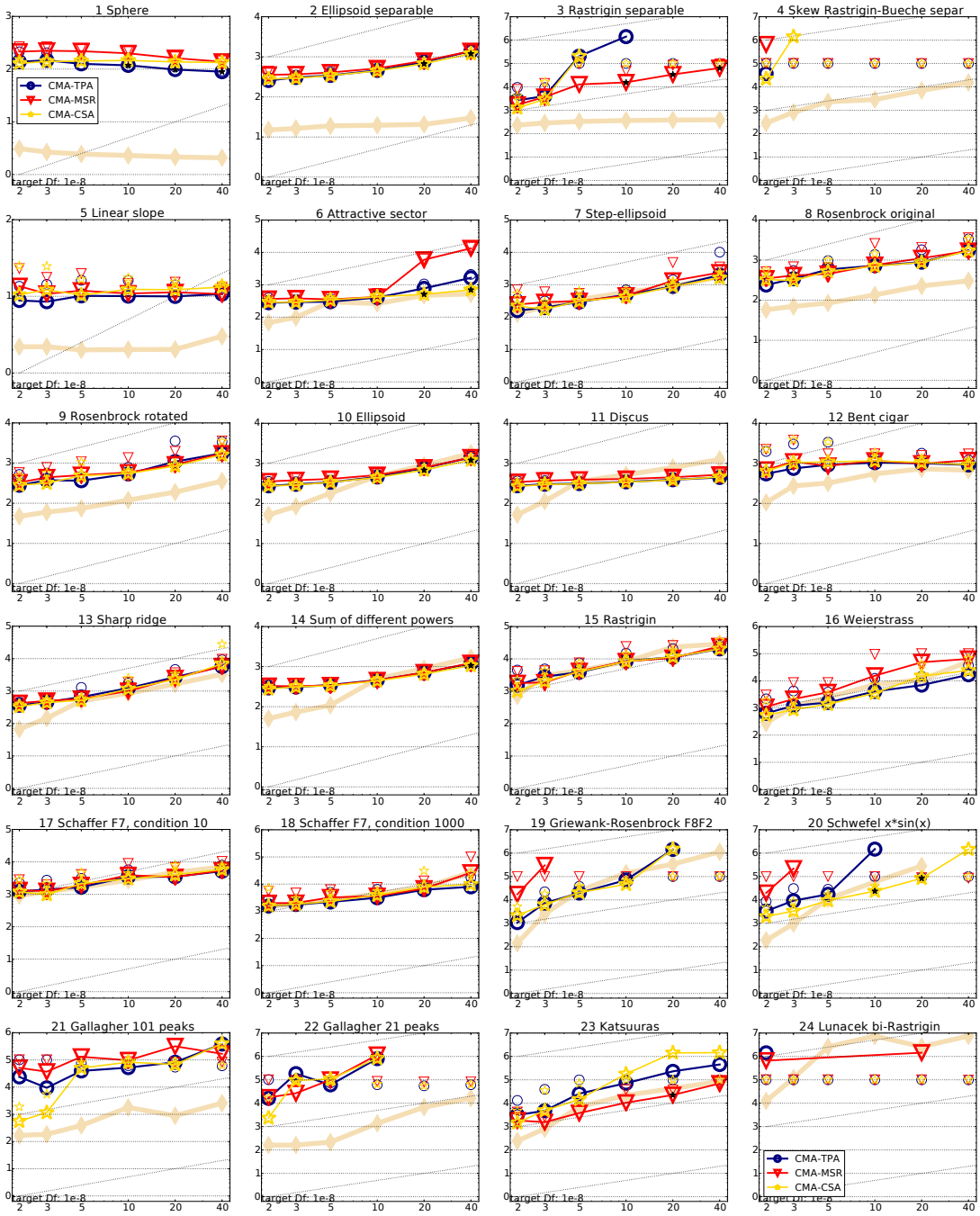

Figure 1: Expected running time (ERT in number of f-evaluations as $\log_{10}$ value), divided by dimension for target function value 10^{-8} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○:CMA-TPA, ▽:CMA-MSR, ⋆:CMA-CSA

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	12	15/15
CMA-TPA	**3.2**(2)	**9.2**(4)	**14**(6)	**20**(5)	**24**(4)	**36**(4)	**47**(8)	15/15
CMA-MSR	3.6(4)	12(3)	21(5)	31(2)	41(6)	62(9)	82(6)	15/15
CMA-CSA	3.8(3)	10(4)	16(3)	22(4)	28(3)	40(3)	52(5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f2	83	87	88	89	90	92	94	15/15
CMA-TPA	**10**(2)	**12**(2)	14(0.5)	15(3)	15(2)	17(3)	18(3)	15/15
CMA-MSR	12(3)	13(2)	14(2)	15(3)	16(2)	18(2)	20(1)	15/15
CMA-CSA	11(2)	13(2)	**14**(1)	**14**(2)	**15**(1)	**16**(1)	**17**(2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f3	716	1622	1637	1642	1646	1650	1654	15/15
CMA-TPA	**0.81**(0.7)	9.3(10)	632(925)	630(1153)	629(926)	628(766)	627(458)	5/15
CMA-MSR	1.7(2)	**5.7**(2)	**36**(86)	**36**(154)	**36**(155)	**37**(164)	**38**(12)	14/15
CMA-CSA	1.4(1)	32(82)	623(1075)	622(460)	621(837)	619(840)	618(607)	5/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f4	809	1633	1688	1758	1817	1886	1903	15/15
CMA-TPA	2.7(4)	∞	∞	∞	∞	∞	∞ *5e5*	0/15
CMA-MSR	2.2(1)	∞	∞	∞	∞	∞	∞ *5e5*	0/15
CMA-CSA	**2.2**(3)	∞	∞	∞	∞	∞	∞ *5e5*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f5	10	10	10	10	10	10	10	15/15
CMA-TPA	4.0(2)	**5.0**(2)	**5.1**(2)	**5.1**(2)	**5.1**(2)	**5.1**(2)	**5.1**(2)	15/15
CMA-MSR	4.2(2)	5.8(3)	5.9(2)	5.9(3)	5.9(2)	5.9(2)	5.9(2)	15/15
CMA-CSA	**3.6**(0.9)	5.0(2)	5.2(2)	5.2(2)	5.2(2)	5.2(3)	5.2(2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f6	114	214	281	404	580	1038	1332	15/15
CMA-TPA	2.2(0.9)	1.9(0.2)	**1.9**(0.7)	**1.7**(0.5)	**1.4**(0.3)	**1.0**(0.2)	**1.0**(0.1)	15/15
CMA-MSR	2.5(0.6)	2.0(0.5)	2.1(0.4)	1.9(0.3)	1.6(0.1)	1.2(0.1)	1.2(0.2)	15/15
CMA-CSA	**2.0**(0.9)	**1.9**(0.3)	2.0(0.4)	1.8(0.1)	1.5(0.3)	1.2(0.2)	1.1(0.2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f7	24	324	1171	1451	1572	1572	1597	15/15
CMA-TPA	**4.1**(2)	**0.98**(1)	0.93(0.5)	0.86(0.4)	0.82(0.3)	0.82(0.3)	**0.83**(0.7)	15/15
CMA-MSR	5.3(5)	1.1(0.7)	0.94(0.4)	0.90(0.4)	0.90(0.6)	0.90(0.6)	0.92(0.5)	15/15
CMA-CSA	4.8(2)	1.3(1)	**0.87**(0.8)	**0.80**(0.9)	**0.80**(0.8)	**0.80**(0.7)	0.86(0.6)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f8	73	273	336	372	391	410	422	15/15
CMA-TPA	4.0(2)	6.0(4)	6.1(3)	6.2(2)	6.3(3)	6.5(3)	6.7(3)	15/15
CMA-MSR	4.6(3)	**3.6**(2)	**4.1**(1)	**4.3**(1)	**4.3**(1)	**4.7**(0.7)	**5.1**(0.5)	15/15
CMA-CSA	**3.0**(0.8)	5.1(5)	5.3(4)	5.4(4)	5.5(3)	5.7(2)	6.0(4)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f9	35	127	214	263	300	335	369	15/15
CMA-TPA	**5.4**(2)	**5.8**(3)	**5.2**(2)	**5.0**(1)	**4.8**(1)	**4.9**(1)	**4.8**(0.9)	15/15
CMA-MSR	7.2(0.7)	9.4(7)	7.5(2)	6.8(6)	6.3(3)	6.3(5)	6.4(5)	15/15
CMA-CSA	5.7(0.7)	10(11)	7.7(7)	7.1(4)	6.7(0.5)	6.5(5)	6.4(4)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f10	349	500	574	607	626	829	880	15/15
CMA-TPA	**2.5**(0.4)	2.2(0.2)	2.1(0.2)	2.1(0.2)	2.2(0.1)	1.8(0.1)	1.8(0.1)	15/15
CMA-MSR	2.6(0.6)	**2.1**(0.5)	2.1(0.3)	2.2(0.2)	2.3(0.2)	2.0(0.2)	2.2(0.1)	15/15
CMA-CSA	2.5(0.4)	2.1(0.2)	**2.0**(0.2)	**2.0**(0.1)	**2.1**(0.2)	**1.8**(0.1)	**1.8**(0.1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f11	143	202	763	977	1177	1467	1673	15/15
CMA-TPA	5.1(0.9)	4.6(0.7)	1.3(0.1)	1.1(0.1)	1.0(0.1)	**0.91**(0.1)	0.89(0.1)	15/15
CMA-MSR	5.9(0.7)	5.0(0.3)	1.5(0.2)	1.3(0.2)	1.2(0.1)	1.1(0.1)	1.1(0.1)	15/15
CMA-CSA	**4.9**(1.0)	**4.3**(0.6)	**1.3**(0.2)	**1.1**(0.2)	**1.00**(0.1)	0.91(0.1)	**0.88**(0.1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f12	108	268	371	413	461	1303	1494	15/15
CMA-TPA	8.3(5)	6.1(8)	6.0(3)	6.2(8)	6.2(1)	2.7(3)	2.9(3)	15/15
CMA-MSR	**7.7**(4)	**5.4**(6)	**5.5**(2)	**5.8**(3)	**6.0**(4)	**2.7**(1)	**2.8**(1)	15/15
CMA-CSA	10(12)	7.1(7)	6.9(9)	7.2(8)	7.4(7)	3.5(1)	3.5(5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f13	132	195	250	319	1310	1752	2255	15/15
CMA-TPA	**2.9**(0.9)	3.8(1)	4.2(2)	4.0(1)	1.2(0.2)	1.3(0.4)	1.2(0.2)	15/15
CMA-MSR	3.2(0.5)	3.6(0.7)	**3.8**(0.7)	4.0(0.7)	1.2(0.1)	1.2(0.1)	1.1(0.1)	15/15
CMA-CSA	3.3(0.8)	**3.4**(2)	4.1(1)	**3.9**(0.9)	**1.1**(0.2)	**1.1**(0.2)	**1.1**(0.2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f14	10	41	58	90	139	251	476	15/15
CMA-TPA	2.1(1)	3.3(2)	3.7(1)	3.9(1)	3.9(0.9)	4.0(0.5)	3.1(0.5)	15/15
CMA-MSR	2.5(1)	3.4(2)	4.7(0.7)	5.0(1)	4.4(0.9)	4.1(0.4)	3.1(0.3)	15/15
CMA-CSA	**1.7**(2)	**2.7**(1)	**3.6**(0.8)	**3.7**(0.8)	**3.8**(0.7)	**3.9**(0.6)	**3.0**(0.3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f15	511	9310	19369	19743	20073	20769	21359	15/15
CMA-TPA	1.9(2)	**0.90**(0.5)	**0.87**(0.6)	**0.88**(0.6)	**0.88**(0.7)	**0.88**(0.6)	**0.89**(0.5)	15/15
CMA-MSR	1.9(2)	0.95(0.8)	0.89(0.7)	0.89(0.6)	0.91(0.6)	0.93(0.6)	0.95(0.8)	15/15
CMA-CSA	**1.1**(0.9)	1.1(0.8)	0.91(0.3)	0.92(0.4)	0.92(0.2)	0.92(0.5)	0.92(0.3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f16	120	612	2662	10163	10449	11644	12095	15/15
CMA-TPA	**1.7**(1)	3.1(3)	1.8(1)	0.56(0.3)	0.62(0.8)	0.62(0.6)	0.65(0.3)	15/15
CMA-MSR	5.9(7)	5.8(5)	4.7(4)	1.6(1)	1.6(1)	1.5(2)	1.5(2)	15/15
CMA-CSA	2.2(1)	**1.9**(1)	**1.4**(1)	**0.49**(0.3)	**0.54**(0.2)	**0.55**(0.3)	**0.56**(0.3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f17	5.2	215	899	2861	3669	6351	7934	15/15
CMA-TPA	24(3)	2.6(2)	1.6(2)	0.97(0.4)	0.94(0.3)	**0.88**(0.3)	**1.0**(0.6)	15/15
CMA-MSR	**4.2**(2)	**0.93**(0.2)	0.97(0.6)	**0.83**(0.6)	**0.82**(0.5)	0.96(0.8)	1.1(0.5)	15/15
CMA-CSA	4.2(6)	0.98(0.3)	**0.53**(0.3)	1.0(0.2)	1.2(0.5)	1.1(0.6)	1.3(0.4)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f18	103	378	3968	8451	9280	10905	12469	15/15
CMA-TPA	**0.92**(0.5)	**1.8**(2)	0.67(1)	0.59(0.4)	**0.69**(0.4)	**0.70**(0.3)	**0.85**(0.3)	15/15
CMA-MSR	1.1(0.7)	5.0(6)	1.0(2)	0.70(0.3)	1.0(0.8)	1.2(0.8)	1.3(1)	15/15
CMA-CSA	1.3(2)	2.4(0.1)	**0.61**(0.4)	**0.54**(0.6)	0.74(0.5)	0.77(0.4)	0.90(0.8)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f19	1	1	242	1.0e5	1.2e5	1.2e5	1.2e5	15/15
CMA-TPA	25(21)	**959**(777)	**84**(62)	**0.68**(0.7)	**0.78**(0.5)	**0.80**(0.7)	**0.80**(0.6)	15/15
CMA-MSR	31(60)	2573(3243)	306(78)	67(63)	∞	∞	∞ *5e5*	0/15
CMA-CSA	**19**(12)	2971(3103)	153(107)	0.86(0.7)	0.83(0.7)	0.83(0.7)	0.84(0.6)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f20	16	851	38111	51362	54470	54861	55313	15/15
CMA-TPA	3.9(2)	17(17)	2.0(0.5)	1.5(0.5)	1.5(0.5)	1.5(0.7)	1.5(0.9)	15/15
CMA-MSR	4.8(0.8)	1666(2186)	∞	∞	∞	∞	∞ *5e5*	0/15
CMA-CSA	**3.7**(1)	**9.2**(9)	**1.1**(0.5)	**0.83**(0.4)	**0.80**(0.6)	**0.82**(0.5)	**0.84**(0.5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f21	41	1157	1674	1692	1705	1729	1757	14/15
CMA-TPA	2.2(0.5)	88(75)	**116**(213)	**115**(425)	**114**(216)	**113**(332)	**112**(177)	10/15
CMA-MSR	5.3(23)	206(6)	388(196)	384(439)	382(577)	377(491)	371(430)	6/15
CMA-CSA	**1.9**(1)	**55**(150)	119(226)	148(319)	147(155)	145(278)	143(207)	9/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f22	71	386	938	980	1008	1040	1068	14/15
CMA-TPA	**2.5**(6)	223(4)	**323**(820)	**310**(534)	**301**(348)	**292**(409)	**285**(305)	8/15
CMA-MSR	14(13)	457(1052)	531(574)	508(663)	494(951)	479(519)	467(1081)	7/15
CMA-CSA	4.1(1)	**135**(138)	345(479)	426(534)	535(782)	519(629)	507(413)	6/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f23	3.0	518	14249	27890	31654	33030	34256	15/15
CMA-TPA	3.2(2)	16(23)	8.1(4)	4.2(2)	3.8(5)	3.8(8)	3.7(18)	13/15
CMA-MSR	2.5(2)	**3.2**(6)*	**0.91**(0.6)	**0.52**(0.6)	**0.48**(0.4)	**0.51**(0.5)	**0.53**(0.6)	15/15
CMA-CSA	**2.3**(3)	13(15)	4.7(0.8)	2.5(2)	2.2(2)	2.2(2)	2.1(1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f24	1622	2.2e5	6.4e6	9.6e6	9.6e6	1.3e7	1.3e7	3/15
CMA-TPA	**1.3**(2)	**10**(20)	∞	∞	∞	∞	∞ *5e5*	0/15
CMA-MSR	1.3(1)	33(42)	**1.1**(1)	∞	∞	∞	∞ *5e5*	0/15
CMA-CSA	2.0(2)	∞	∞	∞	∞	∞	∞ *5e5*	0/15

Table 1: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimension 5. The ERT and in braces, as dispersion measure, the half difference between 90 and 10%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding best ERT in the first row. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $f_{\text{opt}} + 10^{-8}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of instances.

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	43	15/15
CMA-TPA	**6.4**(1)*	**11**(1)$^{\star 3}$	**15**(2)$^{\star 3}$	**19**(1)$^{\star 4}$	**24**(2)$^{\star 4}$	**32**(2)$^{\star 4}$	**41**(2)$^{\star 4}$	15/15
CMA-MSR	9.2(1)	16(1.0)	23(3)	30(3)	38(3)	53(3)	68(4)	15/15
CMA-CSA	7.7(1)	14(2)	20(1)	26(2)	32(2)	45(3)	57(4)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f2	385	386	387	388	390	391	393	15/15
CMA-TPA	25(3)	30(2)	33(2)	35(1)	36(1)	37(2)	37(1)	15/15
CMA-MSR	27(4)	32(4)	35(2)	36(2)	37(2)	38(3)	39(2)	15/15
CMA-CSA	**23**(2)	**27**(2)*	**29**(0.9)$^{\star 3}$	**30**(1)$^{\star 3}$	**31**(1.0)$^{\star 3}$	**32**(2)$^{\star 3}$	**33**(1)$^{\star 3}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f3	5066	7626	7635	7637	7643	7646	7651	15/15
CMA-TPA	8.8(5)	1756(2177)	∞	∞	∞	∞	∞ *2e6*	0/15
CMA-MSR	**6.4**(1)	**38**(20)$^{\star 3}$	**70**(45)$^{\star 4}$	**73**(56)$^{\star 4}$	**76**(41)$^{\star 4}$	**81**(36)$^{\star 4}$	**86**(68)$^{\star 4}$	15/15
CMA-CSA	10(8)	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f4	4722	7628	7666	7686	7700	7758	1.4e5	9/15
CMA-TPA	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
CMA-MSR	**5792**(3817)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
CMA-CSA	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f5	41	41	41	41	41	41	41	15/15
CMA-TPA	**4.3**(0.9)	**4.9**(2)	**4.9**(1)	**4.9**(0.8)	**4.9**(1)	**4.9**(0.9)	**4.9**(0.8)	15/15
CMA-MSR	5.0(1)	5.5(2)	5.6(0.6)	5.6(1)	5.6(1)	5.6(1)	5.6(0.8)	15/15
CMA-CSA	4.9(1)	5.8(0.9)	6.0(1)	6.0(1)	6.0(1)	6.0(0.9)	6.0(1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f6	1296	2343	3413	4255	5220	6728	8409	15/15
CMA-TPA	1.6(0.4)	1.3(0.2)	1.2(0.3)	1.3(0.3)	1.4(0.3)	1.5(0.4)	1.6(0.5)	15/15
CMA-MSR	**1.5**(0.7)	1.9(2)	2.4(2)	3.9(4)	5.7(4)	11(6)	13(1)	15/15
CMA-CSA	1.6(0.3)	**1.3**(0.2)	**1.1**(0.2)	**1.1**(0.2)	**1.1**(0.1)*	**1.1**(0.1)$^{\star 2}$	**1.1**(0.1)$^{\star 2}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f7	1351	4274	9503	16523	16524	16524	16969	15/15
CMA-TPA	2.1(1)	2.7(0.7)	**1.6**(0.6)	**1.0**(0.4)	**1.0**(0.4)	**1.0**(0.4)	**1.0**(0.4)	15/15
CMA-MSR	2.1(0.7)	4.2(1)	2.4(1)	1.6(0.5)	1.6(2)	1.6(0.3)	1.5(0.6)	15/15
CMA-CSA	**1.7**(1)	**2.3**(1)	1.7(0.6)	1.1(0.4)	1.1(0.3)	1.1(0.4)	1.0(0.3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f8	2039	3871	4040	4148	4219	4371	4484	15/15
CMA-TPA	**3.1**(0.7)	3.5(0.1)	3.8(1)	3.9(1)	3.9(0.2)	3.9(1)	3.9(0.4)	15/15
CMA-MSR	3.6(0.8)	4.6(3)	4.8(3)	4.8(3)	4.8(0.4)	4.8(0.5)	4.9(3)	15/15
CMA-CSA	3.4(0.7)	**3.4**(0.4)	**3.6**(0.2)	**3.7**(0.3)	**3.8**(0.5)	**3.8**(0.2)	**3.8**(0.2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f9	1716	3102	3277	3379	3455	3594	3727	15/15
CMA-TPA	3.8(0.7)	5.5(8)	5.8(0.4)	5.8(8)	5.8(2)	5.8(7)	5.8(1)	15/15
CMA-MSR	**3.8**(0.9)	4.5(0.5)	4.8(3)	4.8(0.4)	4.8(2)	4.8(0.3)	4.8(2)	15/15
CMA-CSA	3.8(0.4)	**4.1**(0.3)	**4.3**(0.3)	**4.4**(0.2)	**4.4**(0.2)	**4.5**(0.2)	**4.5**(0.4)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f10	7413	8661	10735	13641	14920	17073	17476	15/15
CMA-TPA	1.4(0.1)	1.4(0.1)	1.2(0.1)	1.0(0.1)	0.95(0.1)	0.86(0.0)	0.86(0.0)	15/15
CMA-MSR	1.3(0.2)	1.3(0.1)	1.2(0.1)	0.99(0.1)	0.93(0.1)	0.86(0.0)	0.88(0.0)	15/15
CMA-CSA	**1.2**(0.2)	**1.2**(0.1)	**1.0**(0.1)$^{\star 2}$	**0.86**(0.0)*	**0.81**(0.0)*	**0.74**(0.0)*	**0.76**(0.0)$^{\star 2}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f11	1002	2228	6278	8586	9762	12285	14831	15/15
CMA-TPA	**4.5**(0.3)	2.3(0.1)	0.89(0.0)	0.69(0.0)	0.65(0.0)	0.57(0.0)	0.51(0.0)	15/15
CMA-MSR	4.7(0.5)	2.6(0.2)	1.0(0.1)	0.80(0.0)	0.74(0.0)	0.65(0.0)	0.58(0.0)	15/15
CMA-CSA	4.6(0.2)	**2.3**(0.1)	**0.86**(0.0)	**0.67**(0.0)	**0.63**(0.0)*	**0.55**(0.0)	**0.50**(0.0)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f12	1042	1938	2740	3156	4140	12407	13827	15/15
CMA-TPA	3.8(3)	4.1(2)	3.8(2)	3.9(1)	3.3(2)	1.4(0.3)	1.4(0.5)	15/15
CMA-MSR	3.7(4)	**3.3**(2)	**3.5**(1)	**3.6**(2)	**3.2**(2)	**1.3**(0.5)	**1.4**(0.4)	15/15
CMA-CSA	**3.6**(2)	3.5(2)	3.8(2)	3.9(1)	3.5(1)	1.4(0.3)	1.5(0.4)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f13	652	2021	2751	3507	18749	24455	30201	15/15
CMA-TPA	4.7(5)	4.7(3)	5.0(3)	5.4(2)	1.1(0.7)	1.3(0.6)	1.5(0.5)	15/15
CMA-MSR	4.4(4)	**3.3**(4)	4.9(3)	**4.2**(2)	**0.87**(0.4)	**1.0**(0.4)	1.5(0.8)	15/15
CMA-CSA	**3.2**(4)	4.2(3)	**4.0**(3)	4.5(1)	0.93(0.4)	1.1(0.6)	**1.3**(0.7)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f14	75	239	304	451	932	1648	15661	15/15
CMA-TPA	**3.5**(1)	**2.3**(0.6)	**2.8**(0.8)$^{\star 2}$	**3.1**(0.4)*	**2.8**(0.3)	**3.8**(0.4)	0.71(0.1)	15/15
CMA-MSR	4.2(1)	2.8(0.5)	3.4(0.5)	3.6(0.2)	2.9(0.2)	3.9(0.4)	0.73(0.0)	15/15
CMA-CSA	4.2(1)	2.9(0.5)	3.7(0.2)	4.1(0.3)	3.3(0.3)	3.9(0.3)	**0.67**(0.1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f15	30378	1.5e5	3.1e5	3.2e5	3.2e5	4.5e5	4.6e5	15/15
CMA-TPA	0.94(0.2)	1.1(0.2)	0.63(0.1)	0.64(0.2)	0.64(0.2)	0.48(0.0)	0.49(0.2)	15/15
CMA-MSR	0.98(0.3)	**0.95**(0.2)	**0.54**(0.5)	**0.55**(0.1)	**0.56**(0.3)	**0.43**(0.2)	**0.45**(0.3)	15/15
CMA-CSA	**0.83**(0.6)	0.99(0.3)	0.64(0.2)	0.65(0.1)	0.65(0.2)	0.49(0.2)	0.49(0.2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f16	1384	27265	77015	1.4e5	1.9e5	2.0e5	2.2e5	15/15
CMA-TPA	1.2(0.4)	0.78(0.4)	**0.80**(0.4)	**0.67**(0.4)	**0.63**(0.5)	**0.66**(0.4)	**0.62**(0.2)	15/15
CMA-MSR	**0.80**(0.1)$^{\star 2}$	0.84(0.6)	1.1(0.4)	1.3(1)	3.3(6)	4.7(4)	4.3(5)	12/15
CMA-CSA	1.9(0.5)	**0.64**(0.6)	0.84(0.4)	1.2(1)	1.4(0.6)	1.5(1)	1.4(1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f17	63	1030	4005	12242	30677	56288	80472	15/15
CMA-TPA	**2.7**(1)	1.4(0.6)	1.5(0.9)	**0.94**(0.6)	**0.74**(0.6)	**0.71**(0.4)	**0.80**(0.3)	15/15
CMA-MSR	2.7(0.5)	6.5(3)	3.5(2)	1.9(2)	0.97(0.4)	0.88(0.2)	0.81(0.3)	15/15
CMA-CSA	3.0(2)	**1.0**(0.2)	**1.4**(2)	1.2(0.7)	0.74(0.4)	0.88(0.3)	0.88(0.2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f18	621	3972	19561	28555	67569	1.3e5	1.5e5	15/15
CMA-TPA	1.6(3)	1.3(0.8)	**0.77**(0.4)	**0.96**(0.2)	**0.57**(0.3)	**0.58**(0.5)	**0.74**(0.3)	15/15
CMA-MSR	2.8(14)	2.8(2)	1.4(0.4)	2.0(0.8)	1.2(0.5)	0.83(0.3)	0.87(0.3)	15/15
CMA-CSA	**0.96**(0.3)	**0.72**(0.1)	0.81(0.4)	1.1(0.6)	0.83(0.4)	1.1(2)	1.0(0.3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f19	1	1	3.4e5	4.7e6	6.2e6	6.7e6	6.7e6	15/15
CMA-TPA	**177**(45)	**1.9e4**(84266)	(0.8)	1.2(1)	4.7(6)	**4.3**(5)	**4.3**(4)	1/15
CMA-MSR	212(72)	3.5e4(2e4)	2(0.4)	∞	∞	∞	∞ *2e6*	0/15
CMA-CSA	221(81)	3.3e4(923	**0.82**(0.4)	**0.56**(0.4)	**2.4**(2)	4.5(3)	4.5(3)	1/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f20	82	46150	3.1e6	5.5e6	5.5e6	5.6e6	5.6e6	15/15
CMA-TPA	**4.0**(0.7)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
CMA-MSR	5.1(1)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
CMA-CSA	5.0(1)	**2.5**(0.2)$^{\star 4}$	**0.35**(0.1)*	**0.29**(9e-3)*	**0.29**(0.0)*	**0.29**(4e-3)*	**0.30**(0.1)*	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f21	561	6541	14103	14318	14643	15567	17589	15/15
CMA-TPA	63(187)	248(674)	115(240)	114(95)	111(133)	105(157)	93(165)	6/15
CMA-MSR	**24**(86)	278(305)	449(364)	442(533)	433(508)	407(453)	360(361)	3/15
CMA-CSA	113(403)	**159**(110)	**95**(61)	**94**(182)	**92**(59)	**87**(89)	**77**(110)	7/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f22	467	5580	23491	24163	24948	26847	1.3e5	12/15
CMA-TPA	162(11)	216(94)	∞	∞	∞	∞	∞ *1e6*	0/15
CMA-MSR	254(876)	249(632)	∞	∞	∞	∞	∞ *2e6*	0/15
CMA-CSA	**22**(26)	**145**(279)	∞	∞	∞	∞	∞ *1e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f23	3.2	1614	67457	3.7e5	4.9e5	8.1e5	8.4e5	15/15
CMA-TPA	6.5(5)	23(25)	4.8(12)	3.0(3)	9.3(10)	5.6(13)	5.5(5)	5/15
CMA-MSR	6.8(5)	**2.0**(2)$^{\star 2}$	**0.79**(0.5)*	**0.74**(0.3)	**0.73**(0.2)*	**0.49**(0.2)*	**0.51**(0.1)*	15/15
CMA-CSA	**6.1**(5)	93(33)	13(13)	16(18)	58(59)	35(27)	34(53)	1/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f24	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	5.2e7	15/15
CMA-TPA	6.2(8)	3.9(3)	∞	∞	∞	∞	∞ *2e6*	0/15
CMA-MSR	**4.1**(5)	**1.8**(2)	**0.55**(0.7)	**0.55**(0.4)	**0.55**(1)	**0.55**(0.6)	**0.55**(0.5)	1/15
CMA-CSA	6.0(7)	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Table 2: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimension 20. The ERT and in braces, as dispersion measure, the half difference between 90 and 10%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding best ERT in the first row. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $f_{\text{opt}} + 10^{-8}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of instances.

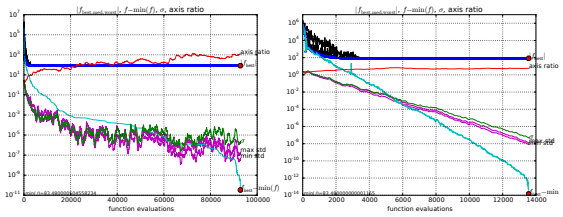

Figure 2: Single runs of IPOP-CMA-MSR (left) and IPOP-CMA-CSA (right) on one instance of the attractive sector function in 20-D. x-axis shows function evaluations. Line with dots (blue): best f-value of the iteration in absolute value, median and worst displayed in thin black lines; cyan line: difference between current f-value and f_{opt}; green line: step-size σ_t, largest and smallest coordinate-wise standard deviation of the sample distribution in purple; red line: square root of the condition number of the covariance matrix.

Acknowledgments

This work was supported by the NumBBO project (grant ANR-2012-MONU-0009) of the French National Research Agency. The author would like to thank Nikolaus Hansen and Anne Auger for their comments on this work.

References

[1] O. Ait Elhara, A. Auger, and N. Hansen. A median success rule for non-elitist evolution strategies: Study of feasibility. In *Genetic and Evolutionary Computation Conference (GECCO 2013), Proceedings*, pages 415–422. ACM, 2013.

[2] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1769–1776. IEEE Press, 2005.

[3] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[4] N. Hansen, D. Arnold, and A. Auger. Evolution strategies. To appear in Janusz Kacprzyk and Witold Pedrycz (Eds.): *Handbook of Computational Intelligence*, Springer.

[5] N. Hansen, A. Atamna, and A. Auger. How to assess step-size adaptation mechanisms in randomised search. In *Parallel Problem Solving from Nature - PPSN XIII, Proceedings*, pages 60–69. Springer, 2014.

[6] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2012: Experimental setup. Technical report, INRIA, 2012.

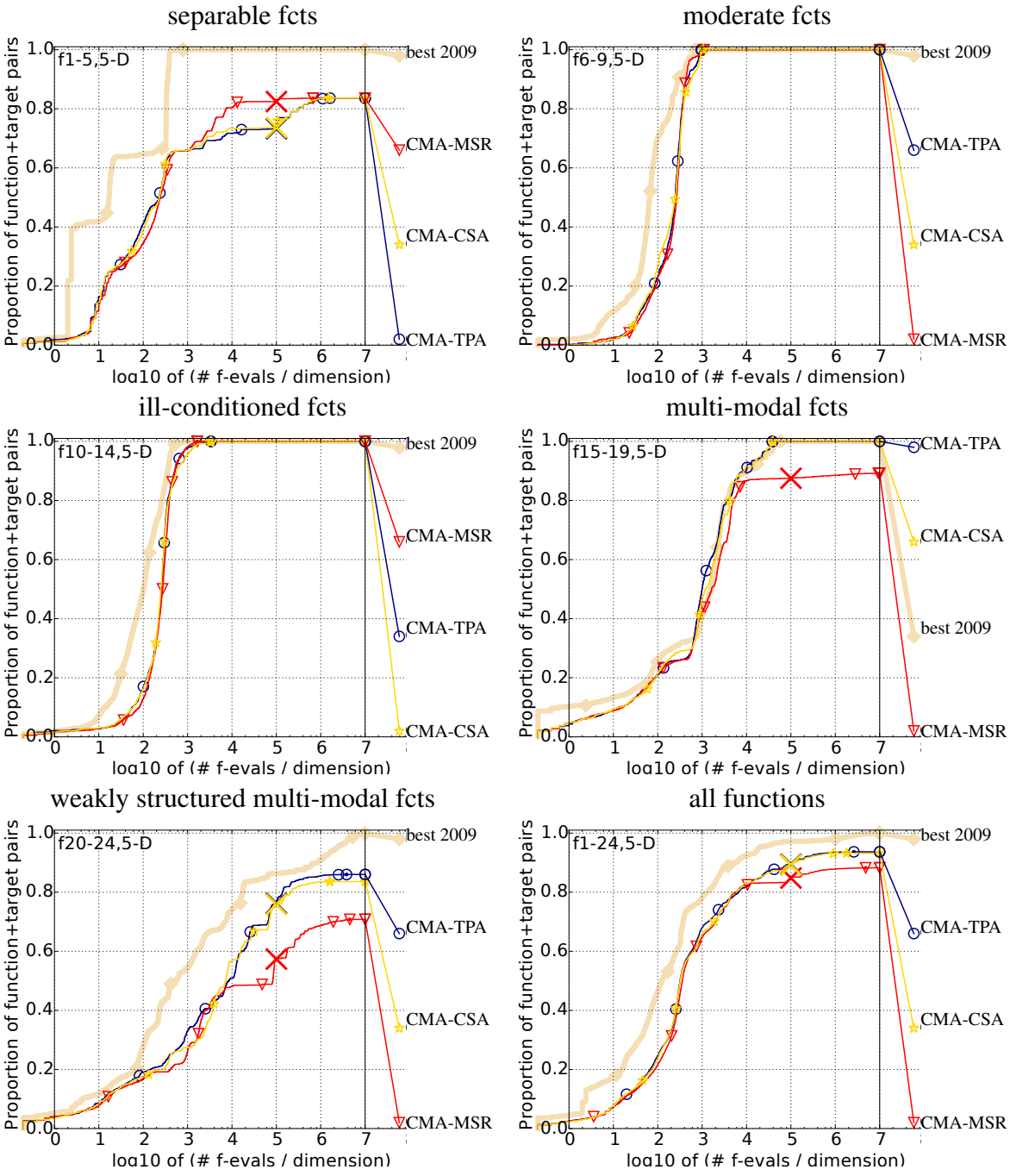

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

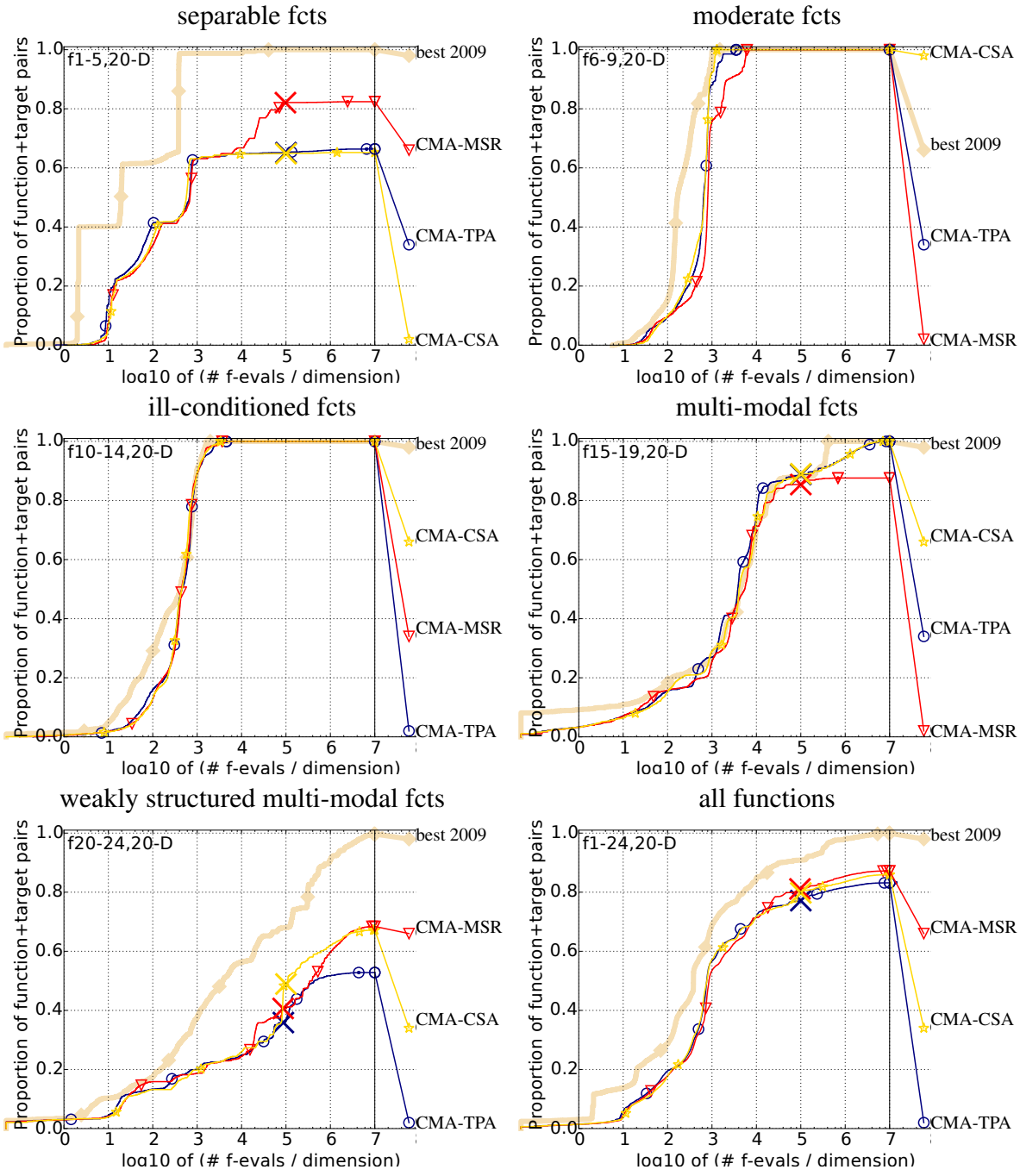

Figure 4: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

[7] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. Updated February 2010.

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

[9] K. Price. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*, pages 153–157, 1997.

[10] I. Rechenberg. *Evolutionsstrategie '94*. Frommann-Holzboog Verlag, 1994.

[11] http://coco.gforge.inria.fr/doku.php?id=bbob-2015-results.

Chapter 6

Markov Chain Analysis of Linear Convergence in Constrained Optimization

We consider in the sequel the following constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ &\text{subject to } g_i(\mathbf{x}) \leq 0,\ i = 1, \cdots, m \ , \end{aligned} \tag{6.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $g_i : \mathbb{R}^n \to \mathbb{R}$ is a linear constraint function of the form $g_i(\mathbf{x}) = \mathrm{b}_i^\intercal \mathbf{x} + c_i$, $\mathrm{b}_i \in \mathbb{R}^n$, $c_i \in \mathbb{R}$, and $m \in \mathbb{N}_>$ is the number of constraints. Without loss of generality, we consider only inequality constraints; indeed, an equality constraint $g_i(\mathbf{x}) = 0$ can be expressed as two inequality constraints $g_i(\mathbf{x}) \leq 0$ and $g_i(\mathbf{x}) \geq 0$. We denote $\mathbf{x}_{\text{opt}}$ the global optimum of (6.1).

To handle the constraints, we use an adaptive augmented Lagrangian approach (see Subsection 4.2.1) where the Lagrange factors and the penalty factors are adapted. In this work, we use one of the following two augmented Lagrangians:

$$h(\mathbf{x}, \gamma, \omega) = f(\mathbf{x}) + \sum_{i=1}^{m} \underbrace{\begin{cases} \gamma^i g_i(\mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x})^2 & \text{if } \gamma^i + \omega^i g_i(\mathbf{x}) \geq 0 \\ -\frac{{\gamma^i}^2}{2\omega^i} & \text{otherwise} \end{cases}}_{\varphi_1(g_i(\mathbf{x}), \gamma^i, \omega^i)} , \tag{6.2}$$

$$h(\mathbf{x}, \gamma, \omega) = f(\mathbf{x}) + \sum_{i=1}^{m} \underbrace{\gamma^i g_i(\mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x})^2}_{\varphi_2(g_i(\mathbf{x}), \gamma^i, \omega^i)} , \tag{6.3}$$

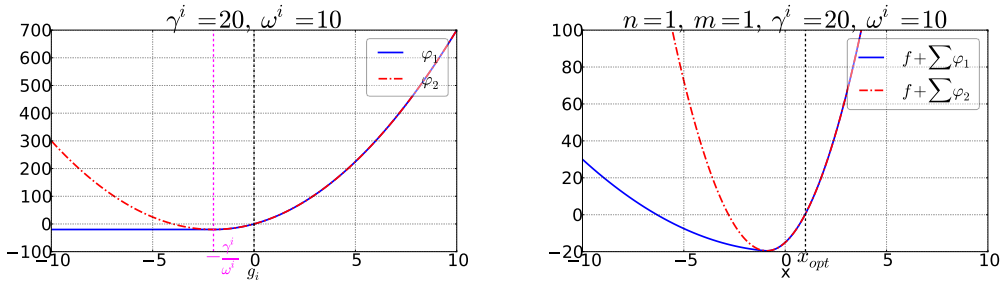

Fig. 6.1 Left: $\varphi_j(g_i(\mathbf{x}), \boldsymbol{\gamma}^i, \boldsymbol{\omega}^i)$ for $j = 1$ (blue) and $j = 2$ (red), as a function of g_i. Right: Augmented Lagrangians, $f(\mathbf{x}) + \sum_{i=1}^{m} \varphi_j(g_i(\mathbf{x}), \boldsymbol{\gamma}^i, \boldsymbol{\omega}^i)$, for $j = 1$ (blue) and $j = 2$ (red), in $n = 1$ with $m = 1$. $f(x) = \frac{1}{2}x^2$, $g_1(x) = x - 1$, and $x_{\text{opt}} = 1$.

where $\boldsymbol{\gamma} = (\gamma^1, \cdots, \gamma^m)^\intercal \in \mathbb{R}^m$ is the vector of Lagrange factors γ^i and $\boldsymbol{\omega} = (\omega^1, \cdots, \omega^m)^\intercal \in (\mathbb{R}^+_>)^m$ is the vector of penalty factors ω^i.

Equation (6.2) defines a practical augmented Lagrangian for the problem in (6.1). The quality of a solution $\mathbf{x}$ is computed by adding either (i) a negative constant $-\frac{\gamma^{i2}}{2\omega^i}$ to $f(\mathbf{x})$ if the solution is "feasible enough" with respect to the constraint g_i ($g_i(\mathbf{x}) < -\frac{\gamma^i}{\omega^i}$, second line in (6.2)) or (ii) $\gamma^i g_i(\mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x})^2$ if $\mathbf{x}$ is "too close" from the constraint boundary (first line in (6.2)), for each constraint g_i. Notice that when the constraints are active at the optimum $\mathbf{x}_{\text{opt}}$, the penalization at $\mathbf{x}_{\text{opt}}$ is 0. This formulation is the recommended choice in practice.

In (6.3), $\gamma^i g_i(\mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x})^2$ is added to the f-value of each individual $\mathbf{x}$. This amounts to considering the case of active constraints. We focus in this work on the case where all the constraints are active, which is the most difficult case in practice. Therefore, we use the augmented Lagrangian in (6.3) in our analysis of convergence. Additionally, this formulation allows us to define a homogeneous Markov chain with the desired properties to deduce linear convergence.

Figure 6.1 shows $\varphi_j(g_i(\mathbf{x}), \boldsymbol{\gamma}^i, \boldsymbol{\omega}^i)$, $j = 1, 2$, as a function of g_i (left) and $h(\mathbf{x}, \boldsymbol{\gamma}, \boldsymbol{\omega})$ in $n = 1$ with $f(x) = \frac{1}{2}x^2$ (sphere function) and $m = 1$ (right). It can be seen that the two augmented Lagrangians in (6.2) and (6.3) are equivalent in the vicinity of the optimum.

To update the Lagrange factors, we use two standard rules given by the literature of augmented Lagrangian methods and presented below:

$$\gamma^i_{t+1} = \max(0, \gamma^i_t + \frac{\omega^i_t}{d_\gamma} g_i(\mathbf{X}_{t+1})) \quad , \tag{6.4}$$

$$\gamma^i_{t+1} = \gamma^i_t + \frac{\omega^i_t}{d_\gamma} g_i(\mathbf{X}_{t+1}) \quad , \tag{6.5}$$

for $i = 1, \cdots, m$. We introduce a damping parameter $d_\gamma > 0$ to moderate the changes in γ^i_t. Equation (6.4) (respectively (6.5)) is used with the augmented Lagrangian in (6.2)

(respectively (6.3)). The update in (6.5) is presented in Subsection 4.2.1. For a detailed discussion on how these updates are derived, see [73].

As for the penalty factors, we use the update originally introduced in [5] for the case of a single constraint and generalize it for the case of multiple constraints.

$$\omega_{t+1}^i = \begin{cases} \omega_t^i \chi^{1/(4d_\omega)} & \text{if } \omega_t^i g_i(\mathbf{X}_{t+1})^2 < k_1 \frac{|h(\mathbf{X}_{t+1},\gamma_t,\omega_t)-h(\mathbf{X}_t,\gamma_t,\omega_t)|}{n} \\ & \text{or } k_2|g_i(\mathbf{X}_{t+1})-g_i(\mathbf{X}_t)| < |g_i(\mathbf{X}_t)| \\ \omega_t^i \chi^{-1/d_\omega} & \text{otherwise} \end{cases} , \; i = 1,\cdots,m \;\; , \tag{6.6}$$

where $\chi, d_\omega, k_1, k_2 \in \mathbb{R}_>^+$. Similarly to the update of the Lagrange factors, we introduce a damping $d_\omega > 0$ to moderate the changes in ω_t^i. This update presents the advantage of increasing the penalty factor only when needed, thereby avoiding an unnecessary ill-conditioning of the problem. As illustrated in (6.6), a penalty factor ω_t^i is increased either if (i) the change in h-value due to the changes in γ_t^i and ω_t^i is smaller than the change in h-value due to the change in $\mathbf{X}_t$ (first inequality in (6.6)). Indeed, we have

$$\omega_t^i g_i(\mathbf{X}_{t+1})^2 \approx |h(\mathbf{X}_{t+1}, \gamma_t + \Delta_i\gamma, \omega_t + \Delta_i\omega) - h(\mathbf{X}_{t+1}, \gamma_t, \omega_t)| \;\; ,$$

where $\Delta_i\gamma = (0,\cdots,\Delta\gamma^i,\cdots,0)^\intercal$ and $\Delta_i\omega = (0,\cdots,\Delta\omega^i,\cdots,0)^\intercal$, and where $\Delta\gamma^i$ and $\Delta\omega^i$ are proportional to $\omega^i g_i(\mathbf{X}_{t+1})$. This aims at preventing premature stagnation [5]. The penalty parameter ω_t^i is also increased if (ii) the change in the constraint value $|g_i(\mathbf{X}_{t+1}) - g_i(\mathbf{X}_t)|$ is significantly smaller than $|g_i(\mathbf{X}_t)|$ (second inequality in (6.6)). By increasing the penalization, we favor solutions near the constraint boundary ($g_i(\mathbf{x}) = 0$). In all other cases, ω_t^i is decreased (second case in (6.6)).

Let us consider a comparison-based adaptive randomized algorithm for unconstrained optimization defined according to (3.2), which we recall below.

$$\mathbf{s}_{t+1} = \mathscr{F}^f(\mathbf{s}_t, \mathbf{U}_{t+1}) \;\; , \tag{6.7}$$

where $\mathscr{F} : \Omega \times (\mathbb{R}^n)^\lambda \to \Omega$ is the transition function of the algorithm, $\mathbf{s}_t$ its state at iteration t, and $\mathbf{U}_{t+1}$ a set of λ i.i.d. random vectors. The superscript f indicates the objective function. Based on this definition, we can easily define an adaptive randomized algorithm with adaptive augmented Lagrangian constraint handling approach from an adaptive randomized algorithm

for unconstrained optimization by taking

$$\mathbf{s}_t{}' = [\mathbf{s}_t, \gamma_t, \omega_t] \ , \tag{6.8}$$

$$h_{(\gamma,\omega)}(\mathbf{x}) := h(\mathbf{x}, \gamma, \omega) \ . \tag{6.9}$$

The new state $\mathbf{s}_{t+1}{}'$ is therefore given by

$$\mathbf{s}_{t+1}{}' = \mathscr{F}^{h_{\gamma_t,\omega_t}}(\mathbf{s}_t{}', \mathbf{U}_{t+1}) \ . \tag{6.10}$$

In this chapter, we present our contributions to constrained optimization. We consider adaptive randomized algorithms defined as in (6.10) where the constraints are handled with an adaptive augmented Lagrangian approach. In Section 6.1, we analyze linear convergence of a $(1+1)$-ES in the case of a single inequality linear constraint using a Markov chain approach. In Section 6.3, we generalize this study to the case of m linear inequality constraints and non-elitist algorithms. In both cases, we show that if the function $\mathscr{D}h_{\bar{\mathbf{x}},\bar{\gamma},\omega} : (\mathbf{x}, \gamma) \mapsto h(\mathbf{x}, \gamma, \omega) - h(\bar{\mathbf{x}}, \bar{\gamma}, \omega)$, with $\bar{\mathbf{x}} \in \mathbb{R}^n$ and $\bar{\gamma} \in \mathbb{R}^m$, is positive homogeneous of degree 2 with respect to $[\bar{\mathbf{x}}, \bar{\gamma}]$, then $\Phi_t = (\mathbf{Y}_t, \Gamma_t, \omega_t)$ is a homogeneous Markov chain, where

$$\mathbf{Y}_t = \frac{\mathbf{X}_t - \bar{\mathbf{x}}}{\sigma_t} \quad \text{and } \Gamma_t = \frac{\gamma_t - \bar{\gamma}}{\sigma_t} \ ,$$

for any $\bar{\mathbf{x}} \in \mathbb{R}^n$ satisfying $g_i(\bar{\mathbf{x}}) = 0$, $i = 1, \cdots, m$, and for any $\bar{\gamma} \in \mathbb{R}^m$. In particular, if the KKT conditions are satisfied for some $\gamma_{\text{opt}} = (\gamma^1_{\text{opt}}, \cdots, \gamma^m_{\text{opt}})^\intercal \in (\mathbb{R}^+)^m$, then Φ_t, with $\bar{\mathbf{x}} = \mathbf{x}_{\text{opt}}$ and $\bar{\gamma} = \gamma_{\text{opt}}$, is a homogeneous Markov chain. We deduce linear convergence under the stability of this Markov chain. In Section 6.2, we show how an adaptive randomized algorithm with an adaptive augmented Lagrangian constraint handling approach can be built from a general adaptive randomized algorithm for unconstrained optimization, in the case of one inequality constraint, and illustrate the proposed methodology on a $(\mu/\mu_W, \lambda)$-CMA-ES.

6.1 Analysis of Linear Convergence of a $(1+1)$-ES with Augmented Lagrangian Constraint Handling

In this work, we analyze linear convergence of an ES for constrained optimization. The algorithm under investigation was introduced in [5] for the case of one inequality constraint. It consists in a $(1+1)$-ES with an augmented Lagrangian constraint handling approach and was—to the best of our knowledge—the first adaptive randomized algorithm observed to

converge linearly on linearly constrained convex quadratic functions (sphere and moderately ill-conditioned ellipsoid), without the need to adapt the covariance matrix.

In an attempt to explain theoretically the observed linear convergence, we use a Markov chain approach to exhibit a homogeneous Markov chain whose stability leads to linear convergence. To obtain the desired Markov chain, we make two modifications to the original algorithm in [5] thereby restricting our analysis to the most interesting case of an active constraint, that is, when the optimum lies on the boundary of the feasible space.

We present in the sequel a slightly modified version of the original paper [7] published in the proceedings of the Genetic and Evolutionary Computation Conference of 2016. We address a minor error in Algorithm 1 (Lines 11, 12, and 13 were originally missing) and improve the writing of the proof of Theorem 2. In this paper, the Lagrange factor and the penalty factor are denoted λ_t and μ_t respectively.

Analysis of Linear Convergence of a $(1+1)$-ES with Augmented Lagrangian Constraint Handling

Asma Atamna, Anne Auger, and Nikolaus Hansen

Inria*
Centre Saclay–Île-de-France
LRI, Université Paris-Saclay

Abstract

We address the question of linear convergence of evolution strategies on constrained optimization problems. In particular, we analyze a $(1+1)$-ES with an augmented Lagrangian constraint handling approach on functions defined on a continuous domain, subject to a single linear inequality constraint. We identify a class of functions for which it is possible to construct a homogeneous Markov chain whose stability implies linear convergence. This class includes all functions such that the augmented Lagrangian of the problem, centered with respect to its value at the optimum and the corresponding Lagrange multiplier, is positive homogeneous of degree 2 (thus including convex quadratic functions as a particular case). The stability of the constructed Markov chain is empirically investigated on the sphere function and on a moderately ill-conditioned ellipsoid function.

1 Introduction

Linear convergence is central in the study of evolution strategies (ESs). Ideally, we want an ES to converge linearly on the widest possible range of optimization problems. As illustrated in [6] for unconstrained optimization, linear convergence can be derived on scaling-invariant functions by exploiting invariance properties of the algorithm at hand on this class of functions: invariance allows to exhibit a Markov chain whose stability leads to linear convergence. In this context, stability is defined as positivity and Harris-recurrence, and is usually obtained by proving φ-irreducibility, aperiodicity, and the existence of a drift function on a small set [10, 6]. Linear convergence then follows from the application of a Law of Large Numbers (LLN). To see how this methodology is applied in practice, one can refer to [4] where linear convergence is proven for the $(1,\lambda)$-ES with self-adaptation on the sphere

*lastname@lri.fr

function, or [5] where the authors show linear convergence of the $(1+1)$-ES with $1/5$th success rule on the class of positive homogeneous functions. Stability is generally difficult to prove "manually". In an attempt to reduce this difficulty, the authors in [7] propose a set of sufficient conditions for a Markov chain to be irreducible and aperiodic.

Linear convergence is also desired on constrained optimization problems [3]. However, little is known about how it can be achieved. Most theoretical works on ESs in the constrained case deal with linear problems with a single linear constraint, as in [2, 1] where the single-step behavior of the $(1+1)$-ES and the $(1,\lambda)$-σSA-ES is analyzed on the linear function with a single linear constraint. In [3], linearly constrained convex quadratic problems are studied for the first time. The authors present an inequality constraint handling method for the $(1+1)$-ES based on augmented Lagrangian and analyze the single-step behavior of the algorithm on the sphere function with one linear inequality constraint. Based on this analysis, they design an update rule for the penalty parameter of the augmented Lagrangian so that the algorithm is empirically observed to converge linearly on sphere and moderately ill-conditioned ellipsoid problems.

In this work, we go one step further into understanding theoretically how linear convergence can be achieved for ESs implementing an augmented Lagrangian constraint handling approach. We introduce a variant of the algorithm presented in [3] and analyze its behavior on the problem of minimizing a function defined on a continuous domain, subject to a single linear inequality constraint. We show that for objective functions such that the corresponding augmented Lagrangian minus its value at the optimum and the corresponding Lagrange multiplier is positive homogeneous of degree 2, one can construct a homogeneous Markov chain and prove linear convergence assuming its stability. Similarly to the unconstrained case, invariance is a key element for constructing the Markov chain. However, invariance alone is not sufficient and another key element is how the parameters of the augmented Lagrangian are updated. Assuming the Markov chain is stable, we prove linear convergence of the solution at a given iteration towards the optimum of the problem, as well as linear convergence of both the Lagrange factor and the step-size towards the Lagrange multiplier associated to the optimum and zero respectively. Then, we empirically investigate the stability of the constructed Markov chain.

The rest of this paper is organized as follows: we formally define the optimization problem we consider in Section 2 and discuss the augmented Lagrangian method in Section 3. We present our algorithm and discuss its invariance properties in Section 4. In Section 5, we present the Markov chain and prove linear convergence assuming its stability. We present our empirical results in Section 6 and conclude with a discussion on the main result of this paper in Section 7.

1.1 Notations

We define here all the notations which are not explicitly presented in the paper. We denote $\mathbb{R}^+$ the set of positive real numbers and $\mathbb{R}^+_>$ the set of strictly positive real numbers. $\mathbf{x} \in \mathbb{R}^n$ is a column vector, $\mathbf{x}^T$ is its transpose, and $\mathbf{0} \in \mathbb{R}^n$ is the zero vector. $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$, $\sim$ equality in distribution, and $\circ$ the function composition operator. The notation $(1+1)$ represents the "one-plus-one" selection scheme. $\mathbf{I}_{n\times n} \in R^{n\times n}$denotes the identity matrix and $\mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$ the multivariate standard normal distribution. $[\mathbf{x}]_i$ is the ith component of vector $\mathbf{x}$ and $[\mathbf{M}]_{ij}$ is the element in the ith row and jth column of matrix $\mathbf{M}$. The derivative with respect to $\mathbf{x}$ is denoted $\nabla_\mathbf{x}$ and the expectation of a random variable $X \sim \pi$ is denoted E_π. Finally, $\mathbf{1}_{\{A\}}$ returns 1 if A is true and 0 otherwise.

2 Optimization Problem

We consider the problem of minimizing a function $f : \mathbb{R}^n \to \mathbb{R}$, n is the dimension of the search space, subject to one linear constraint $g(\mathbf{x}) \leq 0$, where $g : \mathbb{R}^n \to \mathbb{R}$. More formally, we write

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ subject to } g(\mathbf{x}) = \mathrm{b}^T\mathbf{x} + c \leq 0 \ , \tag{1}$$

where $\mathrm{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. We assume the problem to admit a unique global minimum $\mathbf{x}_{\text{opt}}$ and the constraint to be active at $\mathbf{x}_{\text{opt}}$, that is, $g(\mathbf{x}_{\text{opt}}) = 0$.

We consider throughout this paper an ES based on the so-called augmented Lagrangian approach for handling constraints to seek the minimum of this problem. In the next section, we give general notions about the augmented Lagrangian approach.

Since we consider only minimization problems, we will sometimes refer to the minimum as the optimum in the rest of this paper.

3 Augmented Lagrangian Approach

The augmented Lagrangian approach for handling constraints is a combination of the Karush-Kuhn-Tucker (KKT) and penalty function methods. It was introduced for the first time in [8] and [12]. The KKT method defines first-order optimality conditions, referred to as KKT conditions. It introduces the Lagrangian $\mathscr{L} : \mathbb{R}^{n+1} \to \mathbb{R}$ defined as

$$\mathscr{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x}) \ , \tag{2}$$

$\mathbf{x} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$, for an objective function f subject to one inequality constraint $g(\mathbf{x}) \leq 0$. Given some regularity conditions - or constraint qualifications - are satisfied, if $\mathbf{x}^* \in \mathbb{R}^n$ is a local minimum of the constrained problem such that f and g are continuously differentiable at $\mathbf{x}^*$, then there exists a non-negative constant λ^*, called the Lagrange multiplier, such that $\nabla_\mathbf{x}\mathscr{L}(\mathbf{x}^*, \lambda^*) = \mathbf{0}$, that is, $\mathbf{x}^*$ is a stationary point for $\mathscr{L}(\mathbf{x}, \lambda^*)$ (stationarity KKT condition). Put differently, given the "right" λ, the optimum of the constrained problem is a stationary point of the Lagrangian.

Considering the optimization problem in (1) and ellipsoid functions $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$, where $\mathbf{H} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with diagonal elements $[\mathbf{H}]_{ii} = \alpha^{\frac{i-1}{n-1}}$, $\alpha > 0$, KKT conditions are satisfied for the unique minimum of the problem

$$\mathbf{x}_{\text{opt}} = -\frac{c}{\mathrm{b}^T\mathbf{H}^{-1}\mathrm{b}}\mathbf{H}^{-1}\mathrm{b} \tag{3}$$

and the Lagrange multiplier

$$\lambda_{\text{opt}} = \frac{c}{\mathrm{b}^T\mathbf{H}^{-1}\mathrm{b}} \ . \tag{4}$$

In augmented Lagrangian approaches, the Lagrangian in (2) is combined with a penalty term, resulting in the augmented Lagrangian function h. The motivation for using the augmented Lagrangian is to overcome the shortcomings of quadratic penalty function methods, where the penalty factor needs to tend to infinity to achieve convergence [11]. This results in an ill-conditioned problem.

Different formulations of the augmented Lagrangian are possible depending on the optimization problem at hand. A broader discussion on augmented Lagrangians is provided in [11]. In our optimization problem, the constraint is active at the optimum $\mathbf{x}_{\text{opt}}$. Therefore, we use the following augmented Lagrangian

$$h(\mathbf{x},\lambda,\mu) = f(\mathbf{x}) + \lambda g(\mathbf{x}) + \frac{\mu}{2}g^2(\mathbf{x}) \ , \tag{5}$$

where a quadratic penalty term $\frac{\mu}{2}g^2(\mathbf{x})$ is added to penalize points lying outside the boundary of the constraint, μ is a positive penalty factor. At each iteration, h is minimized with respect to $\mathbf{x}$. The parameters λ and μ are updated in such a way that λ approaches the Lagrange multiplier while μ guides the search towards solutions on the constraint boundary. Note that the optimum $\mathbf{x}_{\text{opt}}$ (which is also a KKT point) satisfies $\nabla_{\mathbf{x}} h(\mathbf{x}_{\text{opt}}, \lambda_{\text{opt}}, \mu) = \mathbf{0}$, for all $\mu \in \mathbb{R}^+_>$, where λ_{opt} is the Lagrange multiplier associated to $\mathbf{x}_{\text{opt}}$.

Figure 1 shows graphs of the penalty function $\frac{\mu}{2}g^2$, the Lagrangian $\mathscr{L}$, and the augmented Lagrangian h associated to the sphere function $f(x) = \frac{1}{2}x^2$ in dimension $n = 1$, with $g(x) = -x + 1$. KKT conditions are satisfied for the optimum $x_{\text{opt}} = 1$ and the Lagrange multiplier $\lambda_{\text{opt}} = 1$. $\mathscr{L}$ and h are plotted for $\lambda = 10, \lambda_{\text{opt}}$, and $\mu = 10$. For $\lambda = \lambda_{\text{opt}}$, the minimum of both $\mathscr{L}$ and h (dashed green and blue graphs) correspond to x_{opt}. However, for $\lambda = 10$, the minimum of $\mathscr{L}$ is different (green graph). By adding a penalty term (red graph) to the Lagrangian, the minimum of the augmented Lagrangian (blue graph) moves closer to x_{opt}.

Remark 1. The augmented Lagrangian in (5) is designed for the very specific case of an active constraint ($g(\mathbf{x}_{\text{opt}}) = 0$). This choice is motivated by theoretical considerations–mainly the construction of a homogeneous Markov chain. Note that for problems where $\mathbf{x}_{\text{opt}}$ is inside the feasible domain, i.e. $g(\mathbf{x}_{\text{opt}}) \neq 0$ and $\lambda_{\text{opt}} = 0$, $\nabla_{\mathbf{x}} h(\mathbf{x}_{\text{opt}}, \lambda_{\text{opt}}, \mu) \neq \mathbf{0}$. Hence, in practice where such an information about the optimum is not provided, the augmented Lagrangian used in [3] is the appropriate choice.

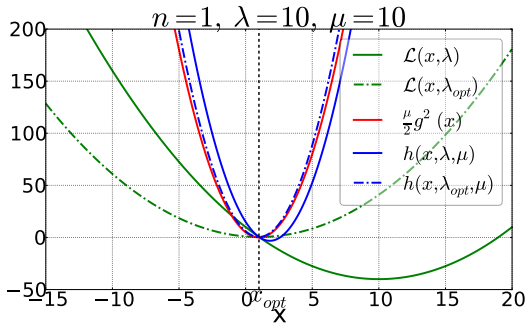

Figure 1: Graphs of $\mathscr{L}(x,\lambda)$ (green), $\mathscr{L}(x,\lambda_{\text{opt}})$ (dashed green), $\frac{\mu}{2}g^2(x)$ (red), $h(x,\lambda,\mu)$ (blue), and $h(x,\lambda_{\text{opt}},\mu)$ (dashed blue) for $\lambda = 10$ and $\mu = 10$ in $n = 1$. $f(x) = \frac{1}{2}x^2$ and $g(x) = -x+1$. $x_{\text{opt}} = 1$ and $\lambda_{\text{opt}} = 1$.

4 Algorithm

In this section, we present a $(1+1)$-ES for solving the optimization problem described in (1), based on the augmented Lagrangian approach described above. The algorithm, summarized in Algorithm 1, iteratively minimizes the augmented Lagrangian function h (5) and adapts the Lagrange and penalty factors λ and μ. It is largely based on the $(1+1)$-ES presented in [3]. Indeed, we use the same update for μ. For λ, however, we modify the update used in [3]. This modification indeed seems to be necessary to be able to exhibit a Markov chain whose stability leads to linear convergence.

Algorithm 1 is a randomized adaptive algorithm. A general randomized adaptive algorithm optimizing a function $f : \mathbb{R}^n \to \mathbb{R}$ subject to a constraint $g(\mathbf{x}) \leq 0$, $g : \mathbb{R}^n \to \mathbb{R}$, is a sequence $(\mathbf{s}_t)_{t\in\mathbb{N}}$ of states, where $\mathbf{s}_t \in \Omega$ is the state of the algorithm at iteration t. The sequence is defined recursively as

$$\mathbf{s}_{t+1} = \mathscr{F}^{(f,g)}(\mathbf{s}_t, \mathbf{U}_{t+1}) \ , \tag{6}$$

where $\mathscr{F}^{(f,g)} : \Omega \times \mathbb{U}^p \to \Omega$ is the transition function of the algorithm and $(\mathbf{U}_{t+1})_{t\in\mathbb{N}}$ is a sequence of independent identically distributed (i.i.d.) random vectors $\mathbf{U}_t \in \mathbb{U}^p$ [6]. For Algorithm 1, the state at iteration t is given by $(\mathbf{X}_t, \sigma_t, \lambda_t, \mu_t)$ where $\mathbf{X}_t \in \mathbb{R}^n$ is the current solution, $\sigma_t \in \mathbb{R}^+$ is the current step-size, $\lambda_t \in \mathbb{R}$ is the current Lagrange factor, and $\mu_t \in \mathbb{R}^+_>$ is the current penalty factor. In fact, Algorithm 1 is based on the $(1+1)$-ES with $1/5$th success rule designed for unconstrained optimization where two additional state variables, λ_t and μ_t, are added to the original state $(\mathbf{X}_t, \sigma_t)$. Indeed, the fitness (the augmented Lagrangian here) in the constrained case is dynamic and is determined by λ_t and μ_t, which are adapted besides $\mathbf{X}_t$ and σ_t.

Given the current state $(\mathbf{X}_t, \sigma_t, \lambda_t, \mu_t)$, a standard normally distributed vector $\mathbf{Z}_{t+1} \in \mathbb{R}^n$ is sampled. It is then multiplied by the step-size σ_t and added to the current solution $\mathbf{X}_t$ to create the first candidate solution $\mathbf{X}^1_{t+1}$, according to Line 3 of Algorithm 1. The second candidate solution is $\mathbf{X}_t$. $\mathbf{X}^1_{t+1}$ and $\mathbf{X}_t$ are then ranked according to their fitness values, where the fitness at iteration t is defined by $h(\mathbf{x}, \lambda_t, \mu_t)$ for a given $\mathbf{x} \in \mathbb{R}^n$. The best point

becomes the solution $\mathbf{X}_{t+1}$ at the next iteration. This is done in Lines 4 and 8 by computing the fitness difference Δh.

The step-size σ_t is adapted with the $1/5$th success rule [9]. It is multiplied by $2^{1/n}$ when $\mathbf{X}^1_{t+1}$ is better than $\mathbf{X}_t$ fitness-wise (Line 9) and by $2^{-1/(4n)}$ otherwise (Line 11). The idea behind this update is to increase (respectively decrease) the step-size if the success probability is larger (respectively smaller) than $1/5$.

The Lagrange factor λ_t is updated (Line 6) if $\mathbf{X}^1_{t+1}$ is accepted: it increases (implying a higher penalization of unfeasible candidate solutions) when $\mathbf{X}^1_{t+1}$ is unfeasible and decreases otherwise. Our update of the Lagrange factor differs from the one in [3] in that it does not restrict λ_t to positive values. This modification appeared to be necessary for us to construct a homogeneous Markov chain whose stability implies linear convergence of the algorithm.

Similarly to the Lagrange factor, the penalty factor μ_t is updated when $\mathbf{X}^1_{t+1}$ is accepted (Line 7), where $\chi, k_1, k_2 \in \mathbb{R}^+_>$. The factor is increased when (i) the penalty term corresponding to $\mathbf{X}^1_{t+1}$ is smaller than the change in h value (first inequality in Line 7). This corresponds to the situation where the Lagrangian part, $f(\mathbf{x}) + \lambda g(\mathbf{x})$, appears to dominate $h(\mathbf{x})$. In this case we increase the penalization so that also the augmenting part, $\mu_t g^2(\mathbf{x})/2$, becomes visible to selection. The other situation where the penalty factor is increased is (ii) when the change in the distance to the constraint boundary $|\Delta g|$ (Line 4) is significantly smaller than the distance to the constraint boundary of the current solution $|g(\mathbf{X}_t)|$ (second inequality in Line 7). In this case, the penalization is increased to avoid premature stagnation when the search process is still far from the constraint boundary, as large values of μ_t guide the search more quickly towards $g(\mathbf{x}) = 0$. When conditions (i) and (ii) are not satisfied, μ_t is decreased to avoid an unnecessary ill-conditioning of the problem.

The updates of $\mathbf{X}_t$ and σ_t depend only on the ranking of h values of the candidate solutions. For λ_t and μ_t however, the algorithm explicitly uses h and g values of $\mathbf{X}_t$ and $\mathbf{X}_{t+1}$.

Referring to (6), the transition function $\mathscr{F}^{(f,g)}$ of Algorithm 1 can be expressed as

$$\begin{aligned}\mathscr{F}^{(f,g)}((\mathbf{X}_t,\sigma_t,\lambda_t,\mu_t),\mathbf{U}_{t+1}) = \big(&\mathscr{G}_1((\mathbf{X}_t,\sigma_t),\varsigma*\mathbf{U}_{t+1}),\mathscr{G}_2(\sigma_t,\varsigma*\mathbf{U}_{t+1}),\\ &\mathscr{G}_3^{(f,g)}(\lambda_t,\mu_t,\mathbf{X}_t,\mathscr{G}_1((\mathbf{X}_t,\sigma_t),\varsigma*\mathbf{U}_{t+1})),\mathscr{G}_4^{(f,g)}(\mu_t,\lambda_t,\mathbf{X}_t,\mathscr{G}_1((\mathbf{X}_t,\sigma_t),\varsigma*\mathbf{U}_{t+1}))\big)\ ,\end{aligned} \tag{7}$$

where $\mathbf{U}_{t+1} = (\mathbf{Z}_{t+1}, \mathbf{0}) \in \mathbb{R}^{n\times 2}$ and

$$\varsigma = Ord(h(\mathbf{X}_t + \sigma_t[\mathbf{U}_{t+1}]_i, \lambda_t, \mu_t)_{i=1,2}) \tag{8}$$

is the permutation of indices of candidate solutions ordered according to h. Where relevant, we will explicitly write the dependence of ς on the variables used to compute candidate solutions and the fitness used to rank them (here, this would read $\varsigma^{h(\mathbf{x},\lambda_t,\mu_t)}_{(\mathbf{X}_t,\sigma_t)}$). The operator $*$ applies the permutation ς to $\mathbf{U}_{t+1}$ and returns the ranked vector $\varsigma*\mathbf{U}_{t+1} = ([\mathbf{U}_{t+1}]_{[\varsigma]_1}, [\mathbf{U}_{t+1}]_{[\varsigma]_2})$. Functions $\mathscr{G}_1$, $\mathscr{G}_2$, $\mathscr{G}_3$, and $\mathscr{G}_4$ compute the new state variables of the algorithm by updating the current state variables $\mathbf{X}_t$, σ_t, λ_t, and μ_t respectively. They are given by

$$\mathbf{X}_{t+1} = \mathscr{G}_1((\mathbf{X}_t,\sigma_t),\varsigma*\mathbf{U}_{t+1}) = \mathbf{X}_t + \sigma_t[\varsigma*\mathbf{U}_{t+1}]_1\ , \tag{9}$$

$$\sigma_{t+1} = \mathscr{G}_2(\sigma_t,\varsigma*\mathbf{U}_{t+1}) = \sigma_t \underbrace{2^{-\frac{1}{4n}+\frac{5}{4n}\mathbf{1}_{\{[\varsigma*\mathbf{U}_{t+1}]_1\neq\mathbf{0}\}}}}_{\eta^*(\varsigma*\mathbf{U}_{t+1})}\ , \tag{10}$$

Algorithm 1 The $(1+1)$-ES with Augmented Lagrangian Constraint Handling

given $n \in \mathbb{N}_>,\ \chi, k_1, k_2 \in \mathbb{R}^+_>$
initialize $\mathbf{X}_0 \in \mathbb{R}^n,\ \sigma_0 \in \mathbb{R}^+_>,\ \lambda_0 \in \mathbb{R},\ \mu_0 \in \mathbb{R}^+_>,\ t = 0$
while not happy
  Compute $\mathbf{X}^1_{t+1} = \mathbf{X}_t + \sigma_t \mathbf{Z}_{t+1}$, where $\mathbf{Z}_{t+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$
  Compute $\Delta g = g(\mathbf{X}^1_{t+1}) - g(\mathbf{X}_t)$ and $\Delta h = h(\mathbf{X}^1_{t+1}, \lambda_t, \mu_t) - h(\mathbf{X}_t, \lambda_t, \mu_t)$
  if $\Delta h \leq 0$ **then**
    $\lambda_{t+1} = \lambda_t + \mu_t g(\mathbf{X}^1_{t+1})$
    $\mu_{t+1} = \begin{cases} \mu_t \chi^{1/4} & \text{if } \mu_t g^2(\mathbf{X}^1_{t+1}) < k_1 \frac{|\Delta h|}{n} \text{ or } k_2 |\Delta g| < |g(\mathbf{X}_t)| \\ \mu_t \chi^{-1} & \text{otherwise} \end{cases}$
    $\mathbf{X}_{t+1} = \mathbf{X}^1_{t+1}$
    $\sigma_{t+1} = \sigma_t 2^{1/n}$
  else
    $\lambda_{t+1} = \lambda_t$
    $\mu_{t+1} = \mu_t$
    $\mathbf{X}_{t+1} = \mathbf{X}_t$
    $\sigma_{t+1} = \sigma_t 2^{-1/(4n)}$
  $t = t + 1$

where $\eta^*(\varsigma * \mathbf{U}_{t+1})$ is the step-size change (we will sometimes omit the dependence on $\varsigma * \mathbf{U}_{t+1}$ for the sake of simplicity),

$$\lambda_{t+1} = \mathscr{G}_3^{(f,g)}(\lambda_t, \mu_t, \mathbf{X}_t, \mathbf{X}_{t+1}) = \lambda_t + \mu_t g(\mathbf{X}_{t+1}) \times \mathbf{1}_{\{[\varsigma * \mathbf{U}_{t+1}]_1 \neq \mathbf{0}\}} \ , \tag{11}$$

$$\mu_{t+1} = \mathscr{G}_4^{(f,g)}(\mu_t, \lambda_t, \mathbf{X}_t, \mathbf{X}_{t+1}) = \begin{cases} \mu_t \beta_t & \text{if } [\varsigma * \mathbf{U}_{t+1}]_1 \neq \mathbf{0} \\ \mu_t & \text{otherwise} \end{cases} \tag{12}$$

with

$$\beta_t = \begin{cases} \chi^{1/4} & \text{if } \mu_t g^2(\mathbf{X}_{t+1}) < k_1 \frac{|h(\mathbf{X}_{t+1}, \lambda_t, \mu_t) - h(\mathbf{X}_t, \lambda_t, \mu_t)|}{n} \\ & \text{or } k_2 |g(\mathbf{X}_{t+1}) - g(\mathbf{X}_t)| < |g(\mathbf{X}_t)| \\ \chi^{-1} & \text{otherwise} \ . \end{cases} \tag{13}$$

4.1 Invariance

We discuss here invariance with respect to transformations of the search space. We distinguish translation-invariance and scale-invariance.

Before giving the formal definitions of translation and scale-invariance, we remind the definition of a group homomorphism.

Definition 1. Let $(G,.)$ and $(H,*)$ be two groups. A function $\Phi : G \to H$ is a group homomorphism if for all $x,y \in G$, $\Phi(x.y) = \Phi(x) * \Phi(y)$.

Let $\mathscr{S}(\Omega)$ be the set of all bijective transformations from the state space Ω to itself and let $\text{Homo}((\mathbb{R}^n,+),(\mathscr{S}(\Omega),\circ))$ (respectively $\text{Homo}((\mathbb{R}^+_>,.),(\mathscr{S}(\Omega),\circ))$) be the set of group homomorphisms from $(\mathbb{R}^n,+)$ (respectively from $(\mathbb{R}^+_>,.)$) to $(\mathscr{S}(\Omega),\circ)$.

Definition 2. A randomized adaptive algorithm with transition function $\mathscr{F}^{(f,g)}$, where f is the objective function being minimized and g is the constraint function, is translation-invariant if there exists a group homomorphism $\Phi \in \text{Homo}((\mathbb{R}^n,+),(\mathscr{S}(\Omega),\circ))$ such that for any objective function f, for any constraint g, for any $\mathbf{x}_0 \in \mathbb{R}^n$, for any state $\mathbf{s} \in \Omega$, and for any $\boldsymbol{u} \in \mathbb{U}^p$,

$$\mathscr{F}^{(f(\mathbf{x}),g(\mathbf{x}))}(\mathbf{s},\boldsymbol{u}) = \Phi(-\mathbf{x}_0)\left(\mathscr{F}^{(f(\mathbf{x}-\mathbf{x}_0),g(\mathbf{x}-\mathbf{x}_0))}(\Phi(\mathbf{x}_0)(\mathbf{s}),\boldsymbol{u})\right) \ .$$

Informally, the previous definition means that if we transform the current state $\mathbf{s}_t$ of the algorithm via $\Phi(\mathbf{x}_0)$, perform one iteration to optimize $f(\mathbf{x}-\mathbf{x}_0)$ subject to $g(\mathbf{x}-\mathbf{x}_0) \leq 0$, and apply the inverse transformation $\Phi(-\mathbf{x}_0)$ to the resulting state, then we will recover the same state $\mathbf{s}_{t+1}$ as when starting from $\mathbf{s}_t$ and performing one iteration of the algorithm to optimize $f(\mathbf{x})$ subject to $g(\mathbf{x})$.

Definition 3. A randomized adaptive algorithm with transition function $\mathscr{F}^{(f,g)}$, where f is the objective function being minimized and g is the constraint, is scale-invariant if there exists a group homomorphism $\Phi \in \text{Homo}((\mathbb{R}^+_>,.),(\mathscr{S}(\Omega),\circ))$ such that for any objective function f, for any constraint g, for any $\alpha > 0$, for any state $\mathbf{s} \in \Omega$, and for any $\boldsymbol{u} \in \mathbb{U}^p$,

$$\mathscr{F}^{(f(\mathbf{x}),g(\mathbf{x}))}(\mathbf{s},\boldsymbol{u}) = \Phi(1/\alpha)\left(\mathscr{F}^{(f(\alpha\mathbf{x}),g(\alpha\mathbf{x}))}(\Phi(\alpha)(\mathbf{s}),\boldsymbol{u})\right) \ .$$

In the sequel, we prove that Algorithm 1 is translation and scale-invariant.

Proposition 1. *Algorithm 1 is translation-invariant and the associated group homomorphism Φ is defined as*

$$\Phi(\boldsymbol{x}_0)(\boldsymbol{x},\sigma,\lambda,\mu) = (\boldsymbol{x}+\boldsymbol{x}_0,\sigma,\lambda,\mu) \ , \tag{14}$$

for all $\boldsymbol{x}_0,\boldsymbol{x} \in \mathbb{R}^n$ and for all $\sigma,\lambda,\mu \in \mathbb{R}$.

Proof. Consider the homomorphism defined in (14) and let $\mathbf{s}_t = (\mathbf{X}_t,\sigma_t,\lambda_t,\mu_t)$ and $\Phi(\mathbf{x}_0)(\mathbf{s}_t) = (\mathbf{X}'_t,\sigma'_t,\lambda'_t,\mu'_t)$. We have

$$h(\mathbf{X}_t+\sigma_t[\mathbf{U}_{t+1}]_i,\lambda_t,\mu_t) = h(\mathbf{X}'_t+\sigma'_t[\mathbf{U}_{t+1}]_i-\mathbf{x}_0,\lambda'_t,\mu'_t) \ ,$$

where $\mathbf{U}_{t+1} = (\mathbf{Z}_{t+1},\mathbf{0})$. Consequently, the same permutation ς is obtained when ranking candidate solutions $\mathbf{X}'_t+\sigma'_t[\mathbf{U}_{t+1}]_i$, $i=1,2$, on $h(\mathbf{x}-\mathbf{x}_0,\lambda,\mu)$ as when ranking candidate

solutions $\mathbf{X}_t + \sigma_t[\mathbf{U}_{t+1}]_i$, $i = 1,2$, on $h(\mathbf{x},\lambda,\mu)$. Therefore, according to (7), $\mathscr{F}^{(f(\mathbf{x}-\mathbf{x}_0),g(\mathbf{x}-\mathbf{x}_0))}(\Phi(\mathbf{x}_0)(\mathbf{s}_t),\mathbf{U}_{t+1})$ writes

$$\begin{aligned}
\mathbf{X}'_{t+1} &= \mathscr{G}_1((\mathbf{X}'_t,\sigma'_t),\varsigma * \mathbf{U}_{t+1}) = \mathscr{G}_1((\mathbf{X}_t,\sigma_t),\varsigma * \mathbf{U}_{t+1}) + \mathbf{x}_0 \ , \\
\sigma'_{t+1} &= \mathscr{G}_2(\sigma'_t,\varsigma * \mathbf{U}_{t+1}) = \mathscr{G}_2(\sigma_t,\varsigma * \mathbf{U}_{t+1}) \ , \\
\lambda'_{t+1} &= \mathscr{G}_3^{(f(\mathbf{x}-\mathbf{x}_0),g(\mathbf{x}-\mathbf{x}_0))}(\lambda'_t,\mu'_t,\mathbf{X}'_t,\mathbf{X}'_{t+1}) = \mathscr{G}_3^{(f(\mathbf{x}),g(\mathbf{x}))}(\lambda_t,\mu_t,\mathbf{X}_t,\mathbf{X}_{t+1}) \ , \\
\mu'_{t+1} &= \mathscr{G}_4^{(f(\mathbf{x}-\mathbf{x}_0),g(\mathbf{x}-\mathbf{x}_0))}(\mu'_t,\lambda'_t,\mathbf{X}'_t,\mathbf{X}'_{t+1}) = \mathscr{G}_4^{(f(\mathbf{x}),g(\mathbf{x}))}(\mu_t,\lambda_t,\mathbf{X}_t,\mathbf{X}_{t+1}) \ .
\end{aligned} \tag{15}$$

We recover $\mathscr{F}^{(f(\mathbf{x}),g(\mathbf{x}))}((\mathbf{X}_t,\sigma_t,\lambda_t,\mu_t),\mathbf{U}_{t+1})$ by applying the inverse transformation $\Phi(-\mathbf{x}_0)$ to $(\mathbf{X}'_{t+1},\sigma'_{t+1},\lambda'_{t+1},\mu'_{t+1})$. □

Proposition 2. *Algorithm 1 is scale-invariant and the associated group homomorphism* Φ *is defined as*

$$\Phi(\alpha)(x,\sigma,\lambda,\mu) = (x/\alpha,\sigma/\alpha,\lambda,\mu) \ , \tag{16}$$

for all $\alpha \in \mathbb{R}^+_>$, *for all* $\boldsymbol{x} \in \mathbb{R}^n$, *and for all* $\sigma,\lambda,\mu \in \mathbb{R}$.

Proof. Let $\mathbf{s}_t = (\mathbf{X}_t,\sigma_t,\lambda_t,\mu_t)$ and $\Phi(\alpha)(\mathbf{s}_t) = (\mathbf{X}'_t,\sigma'_t,\lambda'_t,\mu'_t)$. We use the same idea as in the previous proof to show that the same permutation ς is obtained when ranking candidate solutions $\mathbf{X}'_t + \sigma'_t[\mathbf{U}_{t+1}]_i$, $i = 1,2$, on $h(\alpha\mathbf{x},\lambda,\mu)$ than when ranking candidate solutions $\mathbf{X}_t + \sigma_t[\mathbf{U}_{t+1}]_i$, $i = 1,2$, on $h(\mathbf{x},\lambda,\mu)$. Therefore, according to (7), $\mathscr{F}^{(f(\alpha\mathbf{x}),g(\alpha\mathbf{x}))}(\Phi(\mathbf{s}_t),\mathbf{U}_{t+1})$ writes

$$\mathbf{X}'_{t+1} = \mathscr{G}_1((\mathbf{X}'_t,\sigma'_t),\varsigma * \mathbf{U}_{t+1}) = \frac{1}{\alpha}\mathscr{G}_1((\mathbf{X}_t,\sigma_t),\varsigma * \mathbf{U}_{t+1}) \ , \tag{17}$$

$$\sigma'_{t+1} = \mathscr{G}_2(\sigma'_t,\varsigma * \mathbf{U}_{t+1}) = \frac{1}{\alpha}\mathscr{G}_2(\sigma_t,\varsigma * \mathbf{U}_{t+1}) \ , \tag{18}$$

$$\begin{aligned}
\lambda'_{t+1} &= \mathscr{G}_3^{(f(\alpha\mathbf{x}),g(\alpha\mathbf{x}))}(\lambda'_t,\mu'_t,\mathbf{X}'_t,\mathbf{X}'_{t+1}) = \mathscr{G}_3^{(f(\mathbf{x}),g(\mathbf{x}))}(\lambda_t,\mu_t,\mathbf{X}_t,\mathbf{X}_{t+1}) \ , \\
\mu'_{t+1} &= \mathscr{G}_4^{(f(\alpha\mathbf{x}),g(\alpha\mathbf{x}))}(\mu'_t,\lambda'_t,\mathbf{X}'_t,\mathbf{X}'_{t+1}) = \mathscr{G}_4^{(f(\mathbf{x}),g(\mathbf{x}))}(\mu_t,\lambda_t,\mathbf{X}_t,\mathbf{X}_{t+1}) \ .
\end{aligned}$$

We recover $\mathscr{F}^{(f(\mathbf{x}),g(\mathbf{x}))}(\mathbf{s}_t,\mathbf{U}_{t+1})$ by applying the inverse transformation $\Phi(1/\alpha)$ to $(\mathbf{X}'_{t+1},\sigma'_{t+1},\lambda'_{t+1},\mu'_{t+1})$. □

5 Analysis

In this section, we investigate the behavior of Algorithm 1 on the augmented Lagrangian h. We start by showing that given a particular condition is satisfied by h, we can construct a homogeneous Markov chain from the state variables of the algorithm, by exploiting its invariance properties as well as the updates of λ_t and μ_t. In the second part, we illustrate how the stability of the constructed Markov chain results in linear convergence of $\mathbf{X}_t$ towards the optimum $\mathbf{x}_{\text{opt}}$, as well as linear convergence of λ_t and σ_t towards λ_{opt} and 0 respectively.

5.1 Homogeneous Markov Chain

Before presenting the Markov chain, we extend the definition of positive homogeneity with respect to zero to any vector $\mathbf{x}^*$.

Definition 4. A function $p : X \to Y$ is positive homogeneous of degree $k > 0$ with respect to $\mathbf{x}^* \in X$ if for all $\alpha > 0$ and for all $\mathbf{x} \in X$,

$$p(\mathbf{x}^* + \alpha\mathbf{x}) = \alpha^k p(\mathbf{x}^* + \mathbf{x}) \quad . \tag{19}$$

By taking $\mathbf{x}^* = 0$, we recover the standard definition of positive homogeneity.

Example 1. Our linear constraint function $g(\mathbf{x}) = \mathrm{b}^T\mathbf{x} + c$ is positive homogeneous of degree 1 with respect to any $\mathbf{x}^* \in \mathbb{R}^n$ such that $g(\mathbf{x}^*) = 0$. The sphere function $p_{\text{sphere}}(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*)$ is also positive homogeneous of degree 2 with respect to $\mathbf{x}^*$.

We will now define two random variables, $\mathbf{Y}_t$ and Λ_t, and prove that if the augmented Lagrangian h satisfies the condition stated below in (21), then $(\mathbf{Y}_t, \Lambda_t, \mu_t)_{t\in\mathbb{N}}$ is a Markov chain. For the proof, we use transition and scale-invariance along with the updates of λ_t and μ_t.

Proposition 3. *Consider the $(1+1)$-ES with augmented Lagrangian constraint handling optimizing the augmented Lagrangian h defined in (5). Let $(\boldsymbol{X}_t, \sigma_t, \lambda_t, \mu_t)_{t\in\mathbb{N}}$ be the Markov chain associated to this ES and let $(\boldsymbol{U}_t)_{t\in\mathbb{N}}$ be the sequence of i.i.d. random vectors where $\boldsymbol{U}_{t+1} = (\boldsymbol{Z}_{t+1}, \mathbf{0}) \in \mathbb{R}^{n\times 2}$ and $\boldsymbol{Z}_{t+1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{n\times n})$. Let*

$$\boldsymbol{Y}_t = \frac{\boldsymbol{X}_t - \bar{\boldsymbol{x}}}{\sigma_t} \quad \textit{and} \quad \Lambda_t = \frac{\lambda_t - \bar{\lambda}}{\sigma_t} \quad . \tag{20}$$

Then, if the function $\mathscr{D}h_{\bar{\boldsymbol{x}},\bar{\lambda},\mu} : \mathbb{R}^{n+1} \to \mathbb{R}$ defined as follows

$$\mathscr{D}h_{\bar{\boldsymbol{x}},\bar{\lambda},\mu}(\boldsymbol{x}, \lambda) = h(\boldsymbol{x}, \lambda, \mu) - h(\bar{\boldsymbol{x}}, \bar{\lambda}, \mu) \quad , \tag{21}$$

where $\boldsymbol{x}, \bar{\boldsymbol{x}} \in \mathbb{R}^n$, $\lambda, \bar{\lambda} \in \mathbb{R}$, and $g(\bar{\boldsymbol{x}}) = 0$, is positive homogeneous of degree 2 with respect to $(\bar{\boldsymbol{x}}, \bar{\lambda})$, then $(\boldsymbol{Y}_t, \Lambda_t, \mu_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain defined independently of $(\boldsymbol{X}_t, \sigma_t, \lambda_t, \mu_t)$ as

$$\boldsymbol{Y}_{t+1} = \mathscr{G}_1((\boldsymbol{Y}_t, 1), \varsigma * \boldsymbol{U}_{t+1})/\eta^* \quad , \tag{22}$$

$$\Lambda_{t+1} = (\mathscr{G}_3^{(f(\boldsymbol{x}+\bar{\boldsymbol{x}}), g(\boldsymbol{x}+\bar{\boldsymbol{x}}))}(\Lambda_t + \bar{\lambda}, \mu_t, \boldsymbol{Y}_t, \eta^*\boldsymbol{Y}_{t+1}) - \bar{\lambda})/\eta^* \quad , \tag{23}$$

$$\mu_{t+1} = \mathscr{G}_4^{(f(\boldsymbol{x}+\bar{\boldsymbol{x}}), g(\boldsymbol{x}+\bar{\boldsymbol{x}}))}(\mu_t, \Lambda_t + \bar{\lambda}, \boldsymbol{Y}_t, \eta^*\boldsymbol{Y}_{t+1}) \quad , \tag{24}$$

where $\eta^* = \eta^*(\varsigma * \boldsymbol{U}_{t+1})$ and

$$\varsigma = Ord(h(\boldsymbol{Y}_t + [\boldsymbol{U}_{t+1}]_i + \bar{\boldsymbol{x}}, \Lambda_t + \bar{\lambda}, \mu_t)_{i=1,2}) \quad . \tag{25}$$

Proof. We show that $\mathbf{Y}_{t+1}$, Λ_{t+1}, and μ_{t+1} only depend on $\mathbf{Y}_t$, Λ_t, μ_t and i.i.d. random variables $\mathbf{U}_{t+1}$, and therefore that $(\mathbf{Y}_t, \Lambda_t, \mu_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain.

Given the definitions of $\mathbf{Y}_t$ and Λ_t in Proposition 3, we can write

$$h(\mathbf{X}_t + \sigma_t[\mathbf{U}_{t+1}]_i, \lambda_t, \mu_t) = h(\sigma_t(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i) + \bar{\mathbf{x}}, \sigma_t\Lambda_t + \bar{\lambda}, \mu_t) \ .$$

Consider ranking the elements of the set

$$\{h(\sigma_t(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i) + \bar{\mathbf{x}}, \sigma_t\Lambda_t + \bar{\lambda}, \mu_t)\}_{i=1,2} \ .$$

We obtain the same permutation when ranking the elements of

$$\{\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}(\sigma_t(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i) + \bar{\mathbf{x}}, \sigma_t\Lambda_t + \bar{\lambda})\}_{i=1,2} \ ,$$

where $\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}$ is defined in (21). $\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}$ being positive homogeneous with respect to $(\bar{\mathbf{x}}, \bar{\lambda})$, the ranking is the same on

$$\{\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i + \bar{\mathbf{x}}, \Lambda_t + \bar{\lambda})\}_{i=1,2}$$

and, consequently, on

$$\{h(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i + \bar{\mathbf{x}}, \Lambda_t + \bar{\lambda})\}_{i=1,2} \ .$$

Therefore, the same permutation ς defined in (25) is obtained when ranking the candidate solutions $\mathbf{X}_t + \sigma_t[\mathbf{U}_{t+1}]_i$, $i = 1,2$, on $h(\mathbf{x}, \lambda_t, \mu_t)$ as when ranking the candidate solutions $\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i$ on $h(\mathbf{x} + \bar{\mathbf{x}}, \Lambda_t + \bar{\lambda}, \mu_t)$. It follows that

$$\begin{aligned}\mathbf{Y}_{t+1} = \frac{\mathbf{X}_{t+1} - \bar{\mathbf{x}}}{\sigma_{t+1}} &= \frac{\mathscr{G}_1((\mathbf{X}_t, \sigma_t), \varsigma * \mathbf{U}_{t+1}) - \bar{\mathbf{x}}}{\mathscr{G}_2(\sigma_t, \varsigma * \mathbf{U}_{t+1})} \\ &= \mathscr{G}_1((\mathbf{Y}_t, 1), \varsigma * \mathbf{U}_{t+1})/\eta^* \ , \end{aligned} \tag{26}$$

where we used scale-invariance properties of $\mathscr{G}_1$ and $\mathscr{G}_2$ ((17) and (18)) and translation-invariance property of $\mathscr{G}_1$ in (15).

On the other hand, we have

$$\Lambda_{t+1} = \frac{\lambda_{t+1} - \bar{\lambda}}{\sigma_{t+1}} = \frac{\mathscr{G}_3^{(f(\mathbf{x}),g(\mathbf{x}))}(\lambda_t, \mu_t, \mathbf{X}_t, \mathbf{X}_{t+1}) - \bar{\lambda}}{\mathscr{G}_2(\sigma_t, \varsigma * \mathbf{U}_{t+1})} \ , \tag{27}$$

where $\mathscr{G}_3$ is given in (11). Using scale-invariance of $\mathscr{G}_2$ and positive homogeneity of g with respect to $\bar{\mathbf{x}}$, it follows that

$$\mathscr{G}_3^{(f(\mathbf{x}),g(\mathbf{x}))}(\lambda_t, \mu_t, \mathbf{X}_t, \mathbf{X}_{t+1}) - \bar{\lambda} = \sigma_t\big(\mathscr{G}_3^{(f(\mathbf{x}+\bar{\mathbf{x}}),g(\mathbf{x}+\bar{\mathbf{x}}))}(\Lambda_t + \bar{\lambda}, \mu_t, \mathbf{Y}_t, \eta^*\mathbf{Y}_{t+1}) - \bar{\lambda}\big) \ .$$

Replacing in (27), we obtain (23).

Remark 2. With the update of λ_t used in [3] ($\lambda_{t+1} = \max(0, \lambda_t + \mu_t g(\mathbf{X}_t + \sigma_t\mathbf{Z}_{t+1})$ if $\Delta h \leq 0$, λ_t otherwise), Λ_{t+1} cannot be written as a function of $(\mathbf{Y}_t, \Lambda_t, \mu_t)$. Indeed, because of the max function, one cannot get rid of σ_t.

μ_{t+1} is given in (24). $\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}$ is positive homogeneous of degree 2 with respect to $(\bar{\mathbf{x}},\bar{\lambda})$. Therefore, according to Definition 4 and for $\alpha=\sigma_t$,

$$\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}(\mathbf{X}_{t+1},\lambda_t)=\sigma_t^2\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}(\eta^*\mathbf{Y}_{t+1}+\bar{\mathbf{x}},\Lambda_t+\bar{\lambda}) \tag{28}$$

and

$$\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}(\mathbf{X}_t,\lambda_t)=\sigma_t^2\mathscr{D}h_{\bar{\mathbf{x}},\bar{\lambda},\mu_t}(\mathbf{Y}_t+\bar{\mathbf{x}},\Lambda_t+\bar{\lambda}) \ , \tag{29}$$

where we used (20). Subtracting (29) from (28), we get

$$\begin{aligned}h(\mathbf{X}_{t+1},\lambda_t,\mu_t)-h(\mathbf{X}_t,\lambda_t,\mu_t)=\sigma_t^2\big(&h(\eta^*\mathbf{Y}_{t+1}+\bar{\mathbf{x}},\Lambda_t+\bar{\lambda},\mu_t)\\&-h(\mathbf{Y}_t+\bar{\mathbf{x}},\Lambda_t+\bar{\lambda},\mu_t)\big) \ .\end{aligned} \tag{30}$$

Using (30) and positive homogeneity of g with respect to $\bar{\mathbf{x}}$, we get

$$\beta_t=\begin{cases}\chi^{1/4} & \text{if } \mu_t g^2(\eta^*\mathbf{Y}_{t+1}+\bar{\mathbf{x}})<k_1\frac{|h(\eta^*\mathbf{Y}_{t+1}+\bar{\mathbf{x}},\Lambda_t+\bar{\lambda},\mu_t)-h(\mathbf{Y}_t+\bar{\mathbf{x}},\Lambda_t+\bar{\lambda},\mu_t)|}{n}\\ & \text{or } k_2|g(\eta^*\mathbf{Y}_{t+1}+\bar{\mathbf{x}})-g(\mathbf{Y}_t+\bar{\mathbf{x}})|<|g(\mathbf{Y}_t+\bar{\mathbf{x}})|\\ \chi^{-1} & \text{otherwise} \ ,\end{cases}$$

therefore, (24) follows. □

The result in Proposition 3 is particularly interesting if $\bar{\mathbf{x}}$ and $\bar{\lambda}$ correspond to the optimum of the constrained problem, $\mathbf{x}_{\text{opt}}$, and to the Lagrange multiplier, λ_{opt}, respectively. In this case, one can express the convergence rate of the algorithm towards $\mathbf{x}_{\text{opt}}$ as a function of the homogeneous Markov chain $(\mathbf{Y}_t,\Lambda_t,\mu_t)_{t\in\mathbb{N}}$, where $\mathbf{Y}_t=\frac{\mathbf{X}_t-\mathbf{x}_{\text{opt}}}{\sigma_t}$ and $\Lambda_t=\frac{\lambda_t-\lambda_{\text{opt}}}{\sigma_t}$. The LLN can then be applied to prove linear convergence if the Markov chain satisfies some stability conditions, which are further discussed in Section 5.

Corollary 1. *Let $(X_t,\sigma_t,\lambda_t,\mu_t)_{t\in\mathbb{N}}$ be the Markov chain associated to Algorithm 1 optimizing the augmented Lagrangian h in (5), where f is a convex quadratic function defined as*

$$f(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} \ , \tag{31}$$

with $\boldsymbol{H}\in\mathbb{R}^{n\times n}$ a symmetric positive-definite matrix. Let $\mathbf{Y}_t=\frac{\boldsymbol{X}_t-\boldsymbol{x}_{opt}}{\sigma_t}$ and $\Lambda_t=\frac{\lambda_t-\lambda_{\text{opt}}}{\sigma_t}$, where $\boldsymbol{x}_{opt}$ is the optimum and λ_{opt} is the associated Lagrange multiplier. Then $(\mathbf{Y}_t,\Lambda_t,\mu_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain defined independently of $(\boldsymbol{X}_t,\sigma_t,\lambda_t,\mu_t)$ as in Equations 22, 23, 24, and 25, where $\bar{\boldsymbol{x}}=\boldsymbol{x}_{opt}$ and $\bar{\lambda}=\lambda_{\text{opt}}$.

Before moving to the proof, we remind that for f convex quadratic and g linear, KKT conditions are also sufficient conditions of optimality, that is, a point satisfying KKT conditions is also an optimum of the constrained problem (see Theorem 16.4 in [11]). Since the problem is unimodal, KKT conditions are satisfied only for $\mathbf{x}_{\text{opt}}$ and λ_{opt}, and we have

$$\nabla_{\mathbf{x}}f(\mathbf{x}_{\text{opt}})+\lambda_{\text{opt}}\nabla_{\mathbf{x}}g(\mathbf{x}_{\text{opt}})=\mathbf{0} \ . \tag{32}$$

Proof. We show that for $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}$, $\mathscr{D}h_{\mathbf{x}_{\text{opt}},\lambda_{\text{opt}},\mu}$ in (21) is positive homogeneous of degree 2 with respect to $(\mathbf{x}_{\text{opt}},\lambda_{\text{opt}}) \in \mathbb{R}^{n+1}$ and therefore, by virtue of Proposition 3, $(\mathbf{Y}_t,\Lambda_t,\mu_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain. We have by definition of h

$$h(\mathbf{x}_{\text{opt}}+\alpha\mathbf{x},\lambda_{\text{opt}}+\alpha\lambda,\mu) = \underbrace{f(\mathbf{x}_{\text{opt}}+\alpha\mathbf{x})}_{A} + \underbrace{(\lambda_{\text{opt}}+\alpha\lambda)g(\mathbf{x}_{\text{opt}}+\alpha\mathbf{x})}_{B} + \underbrace{\frac{\mu}{2}g^2(\mathbf{x}_{\text{opt}}+\alpha\mathbf{x})}_{C} \ .$$

Given $\nabla_{\mathbf{x}} f(\mathbf{y}) = \mathbf{y}^T\mathbf{H}$ and $\nabla_{\mathbf{x}} g(\mathbf{y}) = \mathrm{b}^T$, it follows that

$$\begin{aligned}
A &= \alpha^2 f(\mathbf{x}_{\text{opt}}+\mathbf{x}) + (1-\alpha^2)f(\mathbf{x}_{\text{opt}}) + \alpha(1-\alpha)\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}})\mathbf{x} \ , \\
B &= \alpha^2(\lambda_{\text{opt}}+\lambda)g(\mathbf{x}_{\text{opt}}+\mathbf{x}) + \alpha(1-\alpha)\lambda_{\text{opt}}\nabla_{\mathbf{x}} g(\mathbf{x}_{\text{opt}})\mathbf{x} \ , \\
C &= \alpha^2\frac{\mu}{2}g^2(\mathbf{x}_{\text{opt}}+\mathbf{x}) \ .
\end{aligned}$$

Therefore

$$\begin{aligned}
h(\mathbf{x}_{\text{opt}}+\alpha\mathbf{x},\lambda_{\text{opt}}+\alpha\lambda,\mu) = \alpha^2 h(\mathbf{x}_{\text{opt}}+\mathbf{x},\lambda_{\text{opt}}+\lambda,\mu) + (1-\alpha^2)f(\mathbf{x}_{\text{opt}}) \\
+ \alpha(1-\alpha)(\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}}) + \lambda_{\text{opt}}\nabla_{\mathbf{x}} g(\mathbf{x}_{\text{opt}}))\mathbf{x} \ .
\end{aligned}$$

Using (32) and the fact that the constraint g is active at $\mathbf{x}_{\text{opt}}$, implying that $h(\mathbf{x}_{\text{opt}},\lambda_{\text{opt}},\mu) = f(\mathbf{x}_{\text{opt}})$, we get

$$\mathscr{D}h_{\mathbf{x}_{\text{opt}},\lambda_{\text{opt}},\mu}(\mathbf{x}_{\text{opt}}+\alpha\mathbf{x},\lambda_{\text{opt}}+\alpha\lambda) = \alpha^2\mathscr{D}h_{\mathbf{x}_{\text{opt}},\lambda_{\text{opt}},\mu}(\mathbf{x}_{\text{opt}}+\mathbf{x},\lambda_{\text{opt}}+\lambda) \ .$$

□

Figure 2 shows contour lines of $\mathscr{D}h_{x_{\text{opt}},\lambda_{\text{opt}},\mu}(x,\lambda)$ (21), where the augmented Lagrangian h is defined for a particular convex quadratic function, the sphere $f(x) = \frac{1}{2}x^2$, $x \in \mathbb{R}$, and the constraint function $g(x) = -x+1$. The penalty factor $\mu = 1$. In this setting, $x_{\text{opt}} = 1$ and $\lambda_{\text{opt}} = 1$. We can see from the figure that the function is scaling-invariant with respect to $(x_{\text{opt}},\lambda_{\text{opt}})$: if we zoom in around the point $(x_{\text{opt}},\lambda_{\text{opt}})$, we will still see the same contour lines. Algorithm 1 optimizes the function whose values correspond to a horizontal cut in the graph, that is, to a fixed value of λ. The intersection between the horizontal line $\lambda = \lambda_i$ and the blue line corresponds to $\min_x \mathscr{D}h_{x_{\text{opt}},\lambda_{\text{opt}},\mu}(x,\lambda_i,\mu)$ where $\arg\min_x \mathscr{D}h_{x_{\text{opt}},\lambda_{\text{opt}},\mu}(x,\lambda_i,\mu)$ can be read on the x-axis. For $\lambda = \lambda_{\text{opt}}$, the intersection happens in 0 and the corresponding value on the x-axis is $x_{\text{opt}} = 1$.

5.2 Sufficient Conditions for Linear Convergence

Let us consider Algorithm 1 optimizing the augmented Lagrangian h from (5) such that the function $\mathscr{D}h_{\mathbf{x}_{\text{opt}},\lambda_{\text{opt}},\mu_t}$ defined in (21) is positive homogeneous of degree 2 with respect to $(\mathbf{x}_{\text{opt}},\lambda_{\text{opt}})$, where $\mathbf{x}_{\text{opt}}$ is the optimum of the problem and λ_{opt} is the associated Lagrange multiplier. Let $(\mathbf{X}_t,\sigma_t,\lambda_t,\mu_t)_{t\in\mathbb{N}}$ be the Markov chain generated by the algorithm. Under these assumptions, let $(\Phi_t)_{t\in\mathbb{N}}$, with $\Phi_t = (\mathbf{Y}_t,\Lambda_t,\mu_t)$, be the homogeneous Markov chain

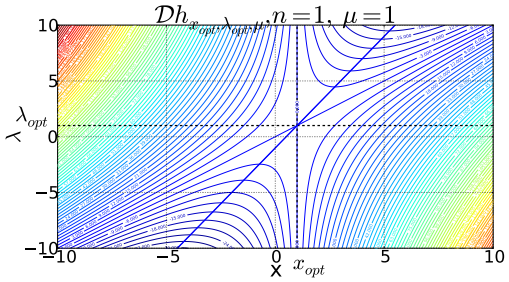

Figure 2: Contour lines of $\mathscr{D}h_{x_{\text{opt}},\lambda_{\text{opt}},\mu}$ for $f(x) = \frac{1}{2}x^2$, $g(\mathbf{x}) = -x+1$, and $\mu = 1$. The vertical (respectively horizontal) dotted black line shows $x_{\text{opt}} = 1$ (respectively $\lambda_{\text{opt}} = 1$). Points where the solid blue line intersects the contour lines represent $\min_x \mathscr{D}h_{x_{\text{opt}},\lambda_{\text{opt}},\mu}(x,\lambda)$ for the corresponding λ.

defined in Proposition 3. The log-progress $\ln \frac{\|\mathbf{X}_{t+1}-\mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_t-\mathbf{x}_{\text{opt}}\|}$ can be expressed as a function of Φ_t as follows

$$\ln \frac{\|\mathbf{X}_{t+1}-\mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_t-\mathbf{x}_{\text{opt}}\|} = \ln \frac{\|\mathbf{Y}_{t+1}\|}{\|\mathbf{Y}_t\|}\eta^*(\varsigma * \mathbf{U}_{t+1}) \ , \tag{33}$$

where ς and η^* are defined in (25) and (10) respectively. By taking the sum then the limit of the average, we obtain the convergence rate

$$\lim_{t\to\infty} \frac{1}{t}\sum_{k=0}^{t-1} \ln \frac{\|\mathbf{X}_{k+1}-\mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_k-\mathbf{x}_{\text{opt}}\|} = \lim_{t\to\infty} \frac{1}{t}\sum_{k=0}^{t-1} \ln \frac{\|\mathbf{Y}_{k+1}\|}{\|\mathbf{Y}_k\|}\eta^*(\varsigma * \mathbf{U}_{k+1}) \ . \tag{34}$$

If the Markov chain $(\Phi_t)_{t\in\mathbb{N}}$ is φ-irreducible and positive Harris-recurrent, then a LLN can be applied to the left-hand side of (34) to show almost sure linear convergence.

Before stating our theorem, we define $\mathscr{R}(\phi)$, $\phi = (\phi_1,\phi_2,\phi_3)$, as the expectation of $\ln \eta^*(\varsigma_{(\phi_1,1)}^{h(\mathbf{x}+\mathbf{x}_{\text{opt}},\phi_2+\lambda_{\text{opt}},\phi_3)} * \mathbf{U})$ for $\mathbf{U} \sim p_{\mathbf{U}}$.

$$\begin{aligned}\mathscr{R}(\phi) &= E\left(\ln \eta^*(\varsigma_{(\phi_1,1)}^{h(\mathbf{x}+\mathbf{x}_{\text{opt}},\phi_2+\lambda_{\text{opt}},\phi_3)} * \mathbf{U})\right) \\ &= \int \ln \eta^*(\varsigma_{(\phi_1,1)}^{h(\mathbf{x}+\mathbf{x}_{\text{opt}},\phi_2+\lambda_{\text{opt}},\phi_3)} * \boldsymbol{u})\ p_{\mathbf{U}}(\boldsymbol{u})d\boldsymbol{u} \ .\end{aligned} \tag{35}$$

We also recall Theorem 17.0.1 from [10], which gives sufficient conditions for the application of the LLN.

Theorem 1 (Theorem 17.0.1 from [10])**.** *Assume that $\boldsymbol{X}$ is a positive Harris-recurrent chain with invariant probability π. Then, the LLN holds for any q such that $\pi(|q|) = \int |q(\boldsymbol{x})|\pi(d\boldsymbol{x}) < \infty$, that is, for any initial state $\boldsymbol{X}_0$, $\lim_{t\to\infty} \frac{1}{t}\sum_{k=0}^{t-1} q(\boldsymbol{X}_k) = \pi(q)$ almost surely.*

Theorem 2. *Let $(\boldsymbol{X}_t,\sigma_t,\lambda_t,\mu_t)_{t\in\mathbb{N}}$ be the Markov chain associated to Algorithm 1 optimizing the augmented Lagrangian h such that the function $\mathscr{D}h_{\boldsymbol{x}_{opt},\lambda_{\text{opt}},\mu_t}$ defined in (21) is positive homogeneous of degree 2 with respect to $(\boldsymbol{x}_{opt},\lambda_{\text{opt}})$ (the optimum and the Lagrange*

multiplier respectively). Let $(\Phi_t)_{t\in\mathbb{N}}$, *with* $\Phi_t = (\boldsymbol{Y}_t, \Lambda_t, \mu_t)$, *be the Markov chain defined in Proposition 3 and assume that it is positive Harris-recurrent with invariant probability measure* π, *that* $E_\pi(|\ln\|[\phi]_1\|\|) < \infty$, $E_\pi(|(\ln|[\phi]_2|)|) < \infty$, *and* $E_\pi(\mathscr{R}(\phi)) < \infty$. *Then, for all* $\boldsymbol{X}_0$, *for all* σ_0, *for all* λ_0, *and for all* μ_0, *linear convergence holds asymptotically almost surely (a.s.), that is*

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{\|\boldsymbol{X}_t - \boldsymbol{x}_{opt}\|}{\|\boldsymbol{X}_0 - \boldsymbol{x}_{opt}\|} = \lim_{t\to\infty}\frac{1}{t}\ln\frac{|\lambda_t - \lambda_{\text{opt}}|}{|\lambda_0 - \lambda_{\text{opt}}|} = \lim_{t\to\infty}\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0} = -CR \text{ a.s.} \quad , \tag{36}$$

where

$$-CR = E_\pi(\mathscr{R}(\phi)) = \int \mathscr{R}(\phi)\pi(d\phi) \quad . \tag{37}$$

Proof. Using the property of the logarithm, we have

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|} = \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{X}_{k+1} - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_k - \mathbf{x}_{\text{opt}}\|} \quad .$$

Then, using (34), we obtain

$$\begin{aligned}\lim_{t\to\infty}\frac{1}{t}\ln\frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|} &= \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{Y}_{k+1}\|}{\|\mathbf{Y}_k\|}\eta^*(\varsigma * \mathbf{U}_{k+1}) \\ &= \lim_{t\to\infty}\left(\frac{1}{t}\sum_{k=0}^{t-1}\ln\|\mathbf{Y}_{k+1}\| - \frac{1}{t}\sum_{k=0}^{t-1}\ln\|\mathbf{Y}_k\| + \frac{1}{t}\sum_{k=0}^{t-1}\ln\eta^*(\varsigma * \mathbf{U}_{k+1})\right) \quad .\end{aligned} \tag{38}$$

$(\Phi_t)_{t\in\mathbb{N}}$ is positive Harris-recurrent with an invariant probability measure π and we have that $E_\pi(|\ln\|[\phi]_1\|\|) < \infty$, $E_\pi(|(\ln|[\phi]_2|)|) < \infty$, and $E_\pi(\mathscr{R}(\phi)) < \infty$. Therefore, we can apply Theorem 1 to the right-hand side of (38). Knowing that $\varsigma = \varsigma_{(\mathbf{Y}_t,1)}^{h(\mathbf{x}+\mathbf{x}_{\text{opt}},\Lambda_t+\lambda_{\text{opt}},\mu_t)}$, it follows

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|} = \int \ln\|[\phi]_1\|\pi(d\phi) - \int \ln\|[\phi]_1\|\pi(d\phi) + \int \mathscr{R}(\phi)\pi(d\phi) = -\text{CR} \quad .$$

The same reasoning applies for $\lim_{t\to\infty}\frac{1}{t}\ln\frac{|\lambda_t-\lambda_{\text{opt}}|}{|\lambda_0-\lambda_{\text{opt}}|}$ and for $\lim_{t\to\infty}\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0}$. Using the property of the logarithm again, we obtain

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{|\lambda_t - \lambda_{\text{opt}}|}{|\lambda_0 - \lambda_{\text{opt}}|} = \lim_{t\to\infty}\left(\frac{1}{t}\sum_{k=0}^{t-1}\ln|\Lambda_{k+1}| - \frac{1}{t}\sum_{k=0}^{t-1}\ln|\Lambda_k| + \frac{1}{t}\sum_{k=0}^{t-1}\ln\eta^*(\varsigma * \mathbf{U}_{k+1})\right) \tag{39}$$

and

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0} = \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\frac{\sigma_{k+1}}{\sigma_k} = \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\eta^*(\varsigma * \mathbf{U}_{k+1}) \quad . \tag{40}$$

By applying the LLN to the right-hand sides of (39) and (40), it follows

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{|\lambda_t - \lambda_{\text{opt}}|}{|\lambda_0 - \lambda_{\text{opt}}|} = \lim_{t\to\infty}\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0} = -\text{CR} \quad .$$

□

6 Empirical Results

By virtue of Corollary 1, all convex quadratic functions satisfy the assumptions in Theorem 1. We consider two of them in our experiments: the sphere function (f_{sphere}) and the ellipsoid function ($f_{\text{ellipsoid}}$), defined in (31) where (i) $\mathbf{H} = \mathbf{I}_{n\times n}$ for f_{sphere} and (ii) $\mathbf{H}$ is a diagonal matrix with diagonal elements $[\mathbf{H}]_{ii} = \alpha^{\frac{i-1}{n-1}}, i = 1,\cdots,n$, for $f_{\text{ellipsoid}}$, with condition number $\alpha = 10$. We choose $\mathrm{b} = (-1,0,\cdots,0)^T$ and $c = 1$ for the linear constraint $g(\mathbf{x}) = \mathrm{b}^T\mathbf{x} + c \leq 0$. According to (3) and (4), KKT conditions are satisfied for the optimum $\mathbf{x}_{\text{opt}} = (1,0,\cdots,0)$ and the Lagrange factor $\lambda_{\text{opt}} = 1$ for both problems.

We run Algorithm 1 and simulate the Markov chain on each problem for different parameter settings in dimensions 10, 50, and 100. We choose $k_1 = 3$, $k_2 = 5$, and $\chi = 2^{1/n}$. For space constraints, we only discuss results obtained in $n = 10$.

6.1 Single Runs

Figure 3 shows single runs of Algorithm 1 on constrained f_{sphere} (left column) and constrained $f_{\text{ellipsoid}}$ (right column) for a moderate initial value of the penalty parameter $\mu_0 = 1$ (first row), a large value $\mu_0 = 10^3$ (second row), and a small value $\mu_0 = 10^{-3}$ (third row). For all runs, $\mathbf{X}_0 = (1,\cdots,1)$, $\sigma_0 = 1$, and $\lambda_0 = 2$. Displayed are the distance to the optimum $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$, the distance to the Lagrange multiplier $|\lambda_t - \lambda_{\text{opt}}|$, the penalty factor μ_t, and the step-size σ_t in log-scale, plotted against the number of iterations.

We observe that the algorithm converges linearly on both f_{sphere} and $f_{\text{ellipsoid}}$ after a certain number of iterations, independently of μ_0. The convergence on $f_{\text{ellipsoid}}$ is slower than on f_{sphere}. In the first case, the initial value $\mu_0 = 1$ is already close to the "stable" value of the penalty parameter and linear convergence of $\mathbf{X}_t$, λ_t, and σ_t towards $\mathbf{x}_{\text{opt}}$, λ_{opt}, and 0 occurs immediately. In the second case, the initial value $\mu_0 = 10^3$ is too large. However, it decreases and converges to a stable value after some iterations. The algorithm then starts to converge linearly. For a too small initial value $\mu_0 = 10^{-3}$, the distance to the optimum (and to the Lagrange multiplier) first decreases, then the algorithm stagnates. The reason is that for small values of μ_t, the Lagrange factor λ_t varies very little (see Line 6 in Algorithm 1), therefore the augmented Lagrangian does not change much between iterations, resulting in stagnation. After some iterations, however, μ_t increases again and eventually converges to a stationary value. Once μ_t is stationary, $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$, $|\lambda_t - \lambda_{\text{opt}}|$, and σ_t start to decrease linearly.

6.2 Simulations of the Markov Chain

Figure 4 shows simulations of the Markov chain $(\mathbf{Y}_t, \Lambda_t, \mu_t)_{t\in\mathbb{N}}$ defined in Proposition 3 on constrained f_{sphere} (left column) and constrained $f_{\text{ellipsoid}}$ (right column), for different initial values of the penalty parameter ($\mu_0 = 1, 10^3, 10^{-3}$ in first, second, and third row respectively). The figure shows the evolution of the normalized distance to $\mathbf{x}_{\text{opt}}$, $\|\mathbf{Y}_t\|$, the normalized distance to λ_{opt}, $|\Lambda_t|$, and the penalty factor, μ_t. We choose $\mathbf{Y}_0 = (1,\cdots,1)$ and $\Lambda_0 = 1$ in all simulations.

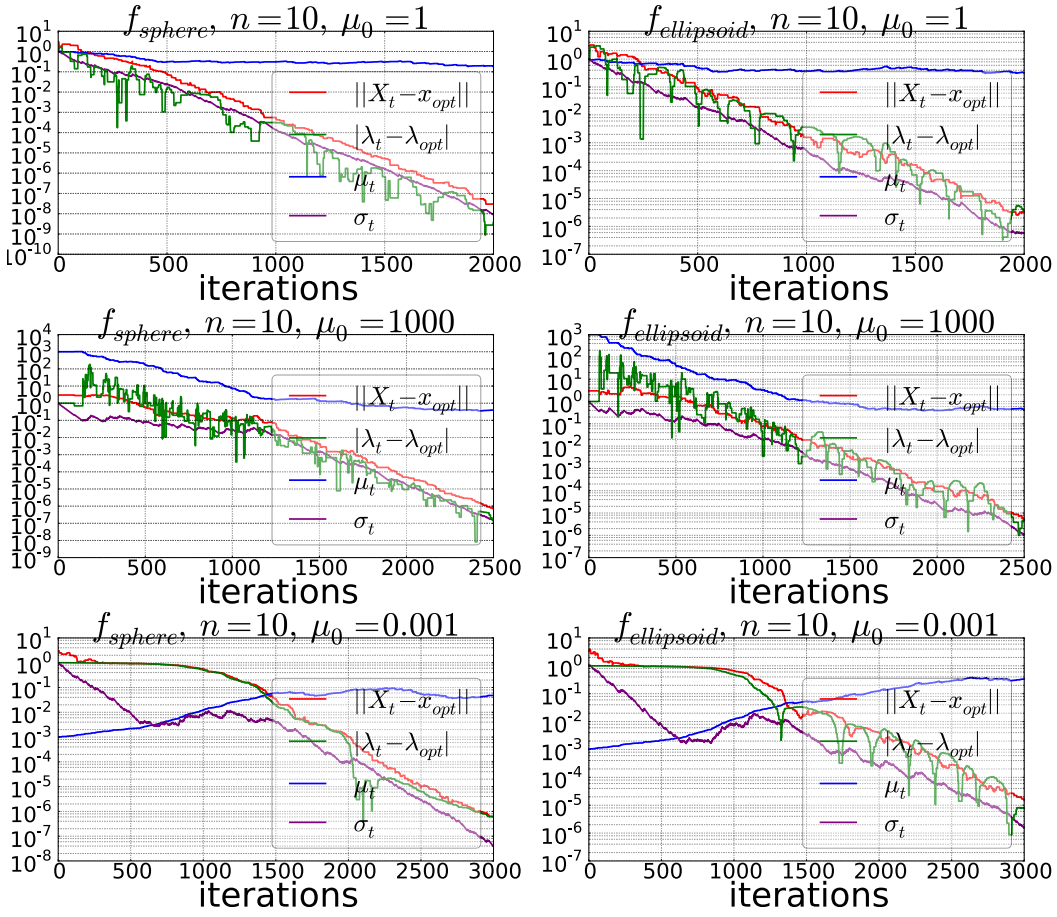

Figure 3: Single runs of the $(1+1)$-ES with augmented Lagrangian constraint handling on constrained f_{sphere} (left column) and constrained $f_{\text{ellipsoid}}$ (right column) for different initial values of μ_t in $n = 10$. Parameters of the constraint g are $\text{b} = (-1, 0, \cdots, 0)^T$ and $c = 1$. $\mathbf{X}_0 = (1, \cdots, 1)^T$ and $\lambda_0 = 2$.

It can be seen from the graphs that the variables of the Markov chain seem to converge to a stationary distribution, even for too small or too large initial values of μ_t. The bump in $\|\mathbf{Y}_t\|$ and $|\Lambda_t|$ graphs we observe on the third row, for both f_{sphere} and $f_{\text{ellipsoid}}$, can be explained by looking at the third row in Figure 3: when μ_t is too small, $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$ and $|\lambda_t - \lambda_{\text{opt}}|$ stagnate while the step-size σ_t decreases, resulting in an increase of $\|\mathbf{Y}_t\|$ and $|\Lambda_t|$. We observe that μ_t oscillates around about 0.1 on constrained f_{sphere} and around about 0.3 for constrained $f_{\text{ellipsoid}}$. These values are comparable to the ones we observe on single runs in Figure 3.

The stability of the Markov chain depends, however, on the parameters of the algorithm. In simulations not shown due to space limitations, we observe instability of the Markov chain, as well as divergence of the algorithm, for $\chi = 2$ with large values of μ_0 in $n = 100$.

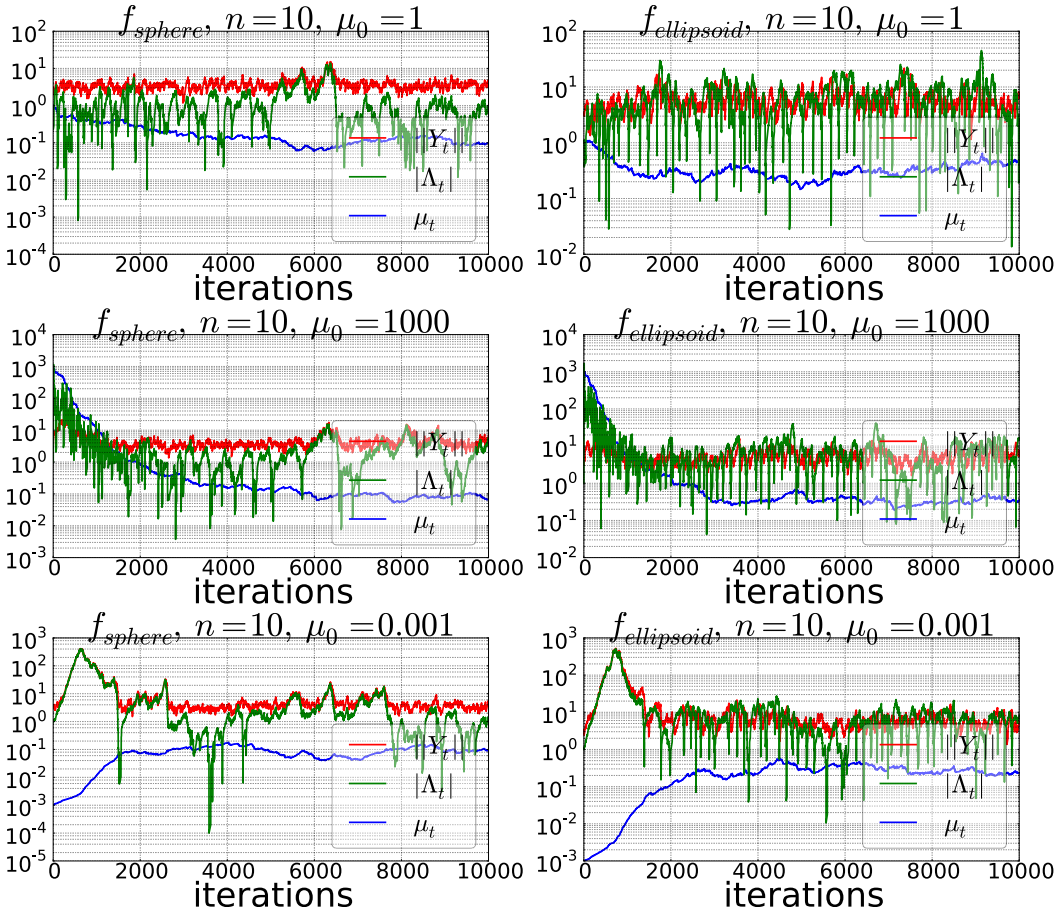

Figure 4: Simulations of the Markov chain $(\mathbf{Y}_t, \Lambda_t, \mu_t)_{t\in\mathbb{N}}$ on constrained f_{sphere} (left column) and constrained $f_{\text{ellipsoid}}$ (right column) for different initial values of μ_t in $n = 10$. Parameters of the constraint g are $\mathrm{b} = (-1, 0, \cdots, 0)^T$ and $c = 1$. $\mathbf{Y}_0 = (1, \cdots, 1)^T$ and $\Lambda_0 = 1$.

7 Discussion

We studied the problem of minimizing a function subject to a single linear constraint. Taking the work of [3] as a starting point, we proposed a $(1+1)$-ES with an augmented Lagrangian constraint handling approach and proved its linear convergence on problems where the associated augmented Lagrangian, minus its value at the optimum and the Lagrange multiplier, is positive homogeneous of degree 2, using a Markov chains approach, and given the stability of the considered Markov chain. To construct the Markov chain, we had to modify the update of the Lagrange factor used in [3] and consider a simpler augmented Lagrangian. Indeed, invariance alone is not sufficient, as the algorithm in [3] is translation and scale-invariant yet we could not find an underlying Markov chain.

Experiments on the constrained sphere and on the moderately ill-conditioned constrained ellipsoid showed stability of the Markov chain, as well as linear convergence of the algorithm, for the discussed parameter settings.

8 Acknowledgments

This work was supported by the grant ANR-2012-MONU-0009 (NumBBO) of the French National Research Agency.

References

[1] D. V. Arnold. On the Behaviour of the $(1, \lambda)$-σSA-ES for a Constrained Linear Problem. In C. A. Coello Coello et al., editors, *Parallel Problem Solving from Nature, PPSN XII*, pages 82–91. Springer, 2012.

[2] 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, 2008.

[3] D. V. Arnold and J. Porter. Towards au Augmented Lagrangian Constraint Handling Approach for the $(1 + 1)$-ES. In *Genetic and Evolutionary Computation Conference*, pages 249–256. ACM Press, 2015.

[4] A. Auger. Convergence Results for the $(1, \lambda)$-SA-ES Using the Theory of φ-Irreducible Markov Chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[5] A. Auger and N. Hansen. Linear Convergence on Positively Homogeneous Functions of a Comparison Based Step-Size Adaptive Randomized Search: the $(1 + 1)$ ES with Generalized One-Fifth Success Rule. Submitted for publication, 2013.

[6] A. Auger and N. Hansen. Linear Convergence of Comparison-Based Step-Size Adaptive Randomized Search via Stability of Markov Chains. *SIAM Journal on Optimization*, 26(3):1589–1624, 2016.

[7] A. Chotard and A. Auger. Verifiable Conditions for Irreducibility, Aperiodicity and T-chain Property of a General Markov Chain. Submitted for publication, 2015.

[8] M. R. Hestenes. Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[9] S. Kern, S. D. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning Probability Distributions in Continuous Evolutionary Algorithms - A Comparative Review. *Natural Computing*, 3(1):77–112, 2004.

[10] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, 1993.

[11] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[12] M. J. D. Powell. A Method for Nonlinear Constraints in Minimization Problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, 1969.

6.2 Augmented Lagrangian Constraint Handling for CMA-ES—Case of a Single Linear Constraint

The following paper [8] presents a general framework for constructing an adaptive randomized algorithm for constrained optimization from an existing adaptive randomized algorithm for unconstrained optimization. We consider the case of a single inequality constraint handled with the adaptive augmented Lagrangian mechanism presented in [5]. To illustrate this general framework, we present a $(\mu/\mu_W, \lambda)$-CMA-ES with median success rule (MSR) step-size adaptation and adaptive augmented Lagrangian constraint handling. The presented algorithm extends the original $(1+1)$-ES in [5] to non-elitist ESs. It can also be modeled as a Markov chain by following a similar approach to the one presented in [7] (see Section 6.1); therefore, its linear convergence can be investigated using the tools from the Markov chain theory. This work was published in the proceedings of the Parallel Problem Solving from Nature conference of 2016.

Augmented Lagrangian Constraint Handling for CMA-ES—Case of a Single Linear Constraint

Asma Atamna, Anne Auger, Nikolaus Hansen

Inria*
LRI (UMR 8623), University of Paris-Saclay, France

Abstract

We consider the problem of minimizing a function f subject to a single inequality constraint $g(\mathbf{x}) \leq 0$, in a black-box scenario. We present a covariance matrix adaptation evolution strategy using an adaptive augmented Lagrangian method to handle the constraint. We show that our algorithm is an instance of a general framework that allows to build an adaptive constraint handling algorithm from a general randomized adaptive algorithm for unconstrained optimization. We assess the performance of our algorithm on a set of linearly constrained functions, including convex quadratic and ill-conditioned functions, and observe linear convergence to the optimum.

1 Introduction

Evolution strategies (ESs) are derivative-free continuous optimization algorithms that are now well-established to solve unconstrained optimization problems of the form $\min_{\mathbf{x}} f(\mathbf{x})$, $f : \mathbb{R}^n \to \mathbb{R}$, where n is the dimension of the search space. The state-of-the-art ES, the covariance matrix adaptation evolution strategy (CMA-ES) [7], is especially powerful at solving a wide range of problems and particularly ill-conditioned problems [8, 5]. It typically exhibits linear convergence. The default CMA-ES algorithm implements comma selection where the best solution is not preserved from one iteration to the next one (contrary to plus selection). Comma selection is an important feature of CMA-ES that entails robustness of the algorithm to various types of ruggedness including noise.

Linear convergence being a central aspect of an ES in the unconstrained case, a $(1+1)$-ES using an adaptive augmented Lagrangian constraint handing—to deal with a single inequality constraint—has been introduced in [3] with the motivation to obtain a linearly converging algorithm. Empirical results show the linear convergence of the algorithm on the sphere and moderately ill-conditioned ellipsoid functions, subject to one linear constraint.

*Research centre Saclay–Île-de-France, TAO team, lastname@lri.fr

In [4], the authors present a variant of the previous $(1+1)$-ES with augmented Lagrangian constraint handling and study theoretically its linear convergence using a Markov chain approach. In both mentioned works, the step-size is adapted using the $1/5$th success rule [10] while the covariance matrix is fixed to the identity. On ill-conditioned problems, however, adapting the covariance matrix is crucial. It is hence natural to wonder whether it is possible to design a CMA-ES variant with augmented Lagrangian constraint handling. The algorithms presented in [3, 4], however, use plus selection and *can thus a priori not be used* directly to design such a variant.

In this context, we consider the constrained problem of minimizing $f : \mathbb{R}^n \to \mathbb{R}$ subject to a single inequality constraint $g(\mathbf{x}) \leq 0$, $g : \mathbb{R}^n \to \mathbb{R}$. More formally, we write

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \quad g(\mathbf{x}) \leq 0 \ . \tag{1}$$

We bring to light that the algorithms previously presented in [3, 4] derive from a more general framework that seamlessly allows to build an adaptive constraint handling algorithm from a general adaptive stochastic search method. We then naturally apply this finding to build a $(\mu/\mu_W, \lambda)$-CMA-ES variant with adaptive augmented Lagrangian constraint handling. We opted for using the median success rule step-size adaptation (MSR) [2] because it is an extension of the $1/5$th success rule algorithm used in [3, 4]. We then test the resulting algorithm—the $(\mu/\mu_W, \lambda)$-MSR-CMA-ES with augmented Lagrangian constraint handling—on a set of functions, including convex quadratic as well as ill-conditioned functions, subject to one linear inequality constraint.

The rest of this paper is organized as follows: we introduce some basics about augmented Lagrangian in Section 2. Then, we define the general framework and apply it to the $(\mu/\mu_W, \lambda)$-MSR-CMA-ES in Section 3. We present our empirical results in Section 4 and conclude with a discussion in Section 5.

Notations We introduce here the notations that are not explicitly defined in the rest of the paper. We denote $\mathbb{R}^+$ the set of positive real numbers and $\mathbb{R}^+_>$ the set of strictly positive real numbers. $\mathbb{N}_>$ is the set of natural numbers without 0. $\mathbf{x} \in \mathbb{R}^n$ is a column vector, $\mathbf{x}^\intercal$ is its transpose, and $\mathbf{0} \in \mathbb{R}^n$ is the zero vector. $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$ and $\sim$ equality in distribution. $(\mu/\mu_W, \lambda)$ denotes comma selection with weighted recombination and $(1+1)$ denotes plus selection with one parent and one offspring. $\mathbf{I}_{n\times n} \in R^{n\times n}$ is the identity matrix. $\mathbf{x}_i$ is the ith component of vector $\mathbf{x}$. The derivative with respect to $\mathbf{x}$ is denoted $\nabla_{\mathbf{x}}$. Finally, $\mathbf{1}_{\{A\}}$ returns 1 if A is true and 0 otherwise.

2 Augmented Lagrangian Methods

Augmented Lagrangian methods are constraint handling approaches that transform the constrained optimization problem into an unconstrained one where an augmented Lagrangian is optimized [9, 12].

The augmented Lagrangian consists of a Lagrangian $\mathscr{L}$ and a penalty function, with $\mathscr{L} : \mathbb{R}^{n+1} \to \mathbb{R}$ defined as

$$\mathscr{L}(\mathbf{x}, \gamma) = f(\mathbf{x}) + \gamma g(\mathbf{x}) \tag{2}$$

for the objective function f subject to one constraint $g(\mathbf{x}) \leq 0$, where $\gamma \in \mathbb{R}$ is the Lagrange factor. The Lagrangian encodes the KKT stationarity condition which states that, given some regularity conditions are satisfied (constraint qualifications), if $\mathbf{x}^* \in \mathbb{R}^n$ is a local minimum of the constrained problem, then there exists a constant $\gamma^* \in \mathbb{R}^+$, called the Lagrange multiplier, such that

$$\underbrace{\nabla_{\mathbf{x}} f(\mathbf{x}^*) + \gamma^* \nabla_{\mathbf{x}} g(\mathbf{x}^*)}_{\nabla_{\mathbf{x}} \mathscr{L}(\mathbf{x}^*, \gamma^*)} = \mathbf{0} \ ,$$

where we assume here that f and g are differentiable at $\mathbf{x}^*$.

A penalty function is combined with the Lagrangian $\mathscr{L}$ to create the augmented Lagrangian h. There exist different ways to construct the augmented Lagrangian and we refer to [11] for a deeper discussion about this topic. In this work, we use the following augmented Lagrangian

$$h(\mathbf{x}, \gamma, \omega) = f(\mathbf{x}) + \begin{cases} \gamma g(\mathbf{x}) + \frac{\omega}{2} g^2(\mathbf{x}) & \text{if } \gamma + \omega g(\mathbf{x}) \geq 0 \\ -\frac{\gamma^2}{2\omega} & \text{otherwise} \end{cases} \ , \tag{3}$$

where $\omega > 0$ is a penalty factor. The same augmented Lagrangian was used for the first time within an ES in [3]. The function h is minimized successively with respect to $\mathbf{x}$, and γ and ω are updated so that γ approaches the Lagrange multiplier γ^* and ω favors feasible solutions. By adapting γ, the penalty factor ω does not have to grow to infinity to achieve convergence, unlike with quadratic penalty function methods [11].

Let $\mathbf{x}_{\text{opt}}$ be the optimum of the constrained problem in (1) and let γ_{opt} be the corresponding Lagrange multiplier. If f and g are differentiable at $\mathbf{x}_{\text{opt}}$, then for all $\omega > 0$,

$$\nabla_{\mathbf{x}} h(\mathbf{x}_{\text{opt}}, \gamma_{\text{opt}}, \omega) = \nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}}) + \max(0, \gamma_{\text{opt}} + \omega g(\mathbf{x}_{\text{opt}})) \nabla_{\mathbf{x}} g(\mathbf{x}_{\text{opt}}) = \mathbf{0} \ .$$

3 A General Framework for Adaptive Augmented Lagrangian Constraint Handling

In [3] and [4], the authors present two $(1+1)$-ESs with an augmented Lagrangian constraint handling approach for the optimization problem in (1). The algorithms derive from a general framework for building a constraint handling adaptive algorithm. This framework starts with a randomized adaptive algorithm for minimizing an unconstrained function $f : \mathbb{R}^n \to \mathbb{R}$: the randomized adaptive algorithm can be identified by the sequence of its states $\mathbf{s}_t$ at iteration t that are iteratively computed from an update function $\mathscr{F}$ such that

$$\mathbf{s}_{t+1} = \mathscr{F}^f(\mathbf{s}_t, \mathbf{U}_{t+1}) \ , \tag{4}$$

where the superscript indicates the function being minimized and where $(\mathbf{U}_t)_{t \in \mathbb{N}_>}$ is a sequence of independent identically distributed (i.i.d.) random vectors. For instance, in the case of a $(1+1)$-ES in [3, 4], the state is a vector of the search space (current estimate of the optimum) and a step-size.

We assume that the state $\mathbf{s}_t$ of the algorithm includes a vector $\mathbf{X}_t \in \mathbb{R}^n$ which typically encodes the current estimate of the optimum at iteration t. Note that the transition function $\mathscr{F}$ above includes a step where candidate solutions are sampled from the current state $\mathbf{s}_t$ and the random vector $\mathbf{U}_{t+1}$, and evaluated on the objective function f.

From the adaptive algorithm above, we construct an algorithm with adaptive constraint handling to take into account a single constraint in the following way: we add to the state of the algorithm two scalars γ_t and ω_t that correspond respectively to the Lagrange factor and the penalty factor of the augmented Lagrangian h at iteration t. Therefore, the state at iteration t is $\mathbf{s}_t{}' = [\mathbf{s}_t, \gamma_t, \omega_t]$. The objective function used at each iteration to evaluate a candidate solution $\mathbf{X}_{t+1}^i$ is now

$$h_{(\gamma_t,\omega_t)}(\mathbf{X}_{t+1}^i) := h(\mathbf{X}_{t+1}^i, \gamma_t, \omega_t) \ , \tag{5}$$

where h is the augmented Lagrangian defined in (3). Finally, the update of the state $\mathbf{s}_t{}'$ of the adaptive algorithm with augmented Lagrangian constraint handling takes place in two steps: first, $\mathbf{s}_t$ is updated via

$$\mathbf{s}_{t+1} = \mathscr{F}^{h_{(\gamma_t,\omega_t)}}(\mathbf{s}_t, \mathbf{U}_{t+1}) \ , \tag{6}$$

where candidate solutions are now evaluated on $h_{(\gamma_t,\omega_t)}$ instead of f. Then, the parameters γ_t and ω_t of h are updated. In [3], γ_t is updated according to

$$\gamma_{t+1} = \max(0, \gamma_t + \omega_t g(\mathbf{X}_{t+1})) \ , \tag{7}$$

while in [4], the authors use the following update

$$\gamma_{t+1} = \gamma_t + \omega_t g(\mathbf{X}_{t+1}) \ . \tag{8}$$

For ω_t, the following update is used in both [3] and [4]

$$\omega_{t+1} = \begin{cases} \omega_t \chi^{1/4} & \text{if } \omega_t g^2(\mathbf{X}_{t+1}) < k_1 \frac{|h(\mathbf{X}_{t+1},\gamma_t,\omega_t) - h(\mathbf{X}_t,\gamma_t,\omega_t)|}{n} \\ & \text{or } k_2 |g(\mathbf{X}_{t+1}) - g(\mathbf{X}_t)| < |g(\mathbf{X}_t)| \\ \omega_t \chi^{-1} & \text{otherwise} \end{cases} \ , \tag{9}$$

for some constants $\chi > 1$, $k_1, k_2 \in \mathbb{R}^+$.

Based on these examples, we introduce some general update functions $\mathscr{G}_\gamma$ and $\mathscr{G}_\omega$ for the updates of γ_t and ω_t defined implicitly via

$$\gamma_{t+1} = \mathscr{G}_\gamma^g((\gamma_t, \omega_t), \mathbf{X}_{t+1}) \tag{10}$$

$$\omega_{t+1} = \mathscr{G}_\omega^{(f,g)}((\mathbf{X}_t, \gamma_t, \omega_t), \mathbf{X}_{t+1}) \ . \tag{11}$$

The superscript in $\mathscr{G}_\gamma$ and $\mathscr{G}_\omega$ indicates that the function value is used in the update.

3.1 The $(\mu/\mu_W, \lambda)$-MSR-CMA-ES with Adaptive Augmented Lagrangian

We now apply the general framework sketched above to the covariance matrix adaptation evolution strategy (CMA-ES) with median success rule step-size adaptation (MSR). We start by presenting the algorithm for the unconstrained case then we give the updates of the augmented Lagrangian parameters γ_t and ω_t.

The (unconstrained) CMA-ES with MSR The original CMA-ES with MSR is given in Algorithm 1, without the highlighted parts. The algorithm proceeds iteratively: at each iteration t, λ candidate solutions (offspring) $\mathbf{X}_{t+1}^i$, $i = 1, \cdots, \lambda$, are sampled according to Line 5, where $\mathbf{X}_t \in \mathbb{R}^n$ is the current estimate of the optimum (mean vector), $\sigma_t \in \mathbb{R}^+$ is the step-size, and $\mathbf{U}_{t+1}^i \in \mathbb{R}^n$, $i = 1, \cdots, \lambda$, are i.i.d. random vectors sampled from the normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{C}_t)$, with mean $\mathbf{0} \in \mathbb{R}^n$ and covariance matrix $\mathbf{C}_t \in \mathbb{R}^{n \times n}$. The offspring are ordered according to their fitness (f-value in the unconstrained case) in Line 6, where $i:\lambda$ is the index of the ith best offspring. The μ best offspring (parents) are then recombined (Line 7) to create the new mean vector $\mathbf{X}_{t+1}$, where the weights $w_i > 0$, $i = 1, \cdots, \mu$, satisfy $w_1 > \cdots > w_\mu$ and $\sum_{i=1}^{\mu} w_i = 1$.

The step-sized σ_t is adapted in Lines 8 to 11 using the MSR step-size adaptation [2]. MSR is a success-based step-size adaptation method which extends the well-known $1/5$th success rule step-size adaptation [10], used with plus selection, to comma selection. The step-size is adapted depending on "success", where the success is defined as the median offspring $\mathbf{X}_{t+1}^{\boldsymbol{m}(\lambda)}$ (fitness-wise) of the current population being better than the jth best offspring $\mathbf{X}_t^{j:\lambda}$ of the previous population. In practice, we choose j to be the 30th percentile–the value for which the median success probability is roughly $1/2$ on the sphere function with optimal step-size [2]. The number K_{succ} of offspring better than $\mathbf{X}_t^{j:\lambda}$ is computed in Line 8. Note that $K_{\text{succ}} \geq \lambda/2$ is equivalent to $h(\mathbf{X}_{t+1}^{\boldsymbol{m}(\lambda)}, \gamma_t, \omega_t) \leq h(\mathbf{X}_t^{j:\lambda}, \gamma_t, \omega_t)$. Therefore, we define the success measure z_t in Line 9 such that $z_t \geq 0$ if and only if $\mathbf{X}_{t+1}^{\boldsymbol{m}(\lambda)}$ is successful. z_t is cumulated in q_{t+1} (Line 10) and, finally, σ_t is updated in Line 11: it increases in the presence of success ($q_{t+1} > 0$) and decreases otherwise in order to increase the probability of success.

The covariance matrix $\mathbf{C}_t$ is adapted with CMA [7] in Lines 12 and 13. The update is a combination of the so-called rank-one update and rank-μ update. A detailed discussion on CMA can be found in [6].

Finally, the jth best offspring is updated in Line 17. Therefore, the state of the algorithm in the unconstrained case is

$$\mathbf{s}_t = (\mathbf{X}_t, \sigma_t, q_t, \boldsymbol{p}_t, \mathbf{C}_t, \mathbf{X}_t^{j:\lambda}) \ .$$

The constrained $(\mu/\mu_W, \lambda)$-MSR-CMA-ES with adaptive augmented Lagrangian As explained in the general framework, the fitness f is replaced with the augmented Lagrangian h in the constrained case. The parameters γ_t and ω_t are adapted in Lines 15 and 16 in Algorithm 1, where changes in comparison to the unconstrained case are highlighted in gray.

The Lagrange factor γ_t is adapted in Line 15. It is increased when the new solution $\mathbf{X}_{t+1}$ is unfeasible and decreased otherwise, unless it is zero. The derivation of this update is discussed in details in [11].

For the penalty parameter ω_t, we use the original update proposed in [3] for the $(1+1)$-ES with augmented Lagrangian. The update rule is given in Line 16. ω_t is increased either when (i) the augmented Lagrangian h does not change "enough" after γ_t and ω_t are updated to avoid stagnation. This is translated by the first inequality where

$$\omega_t g^2(\mathbf{X}_{t+1}) \approx |h(\mathbf{X}_{t+1}, \gamma_t + \Delta\gamma_t, \omega_t + \Delta\omega_t) - h(\mathbf{X}_{t+1}, \gamma_t, \omega_t)|$$

Name	Definition	Name	Definition
$f^{\alpha}_{\text{sphere}}(\mathbf{x})$	$\left(\frac{1}{2}\sum_{i=1}^{n}\mathbf{x}_i^2\right)^{\alpha}$	$f_{\text{diff\_pow}}(\mathbf{x})$	$\sqrt{\sum_{i=1}^{n}\lvert\mathbf{x}_i\rvert^{2+4\frac{i-1}{n-1}}}$
$f_{\text{elli}}(\mathbf{x})$	$\frac{1}{2}\sum_{i=1}^{n}\alpha^{\frac{i-1}{n-1}}\mathbf{x}_i^2$	$f_{\text{rosen}}(\mathbf{x})$	$\sum_{i=1}^{n-1}\left(10^2(\mathbf{x}_i^2-\mathbf{x}_{i+1})^2+(\mathbf{x}_i-1)^2\right)$

Table 1: Definitions of the tested functions, where $f_{\text{sphere}} := f^1_{\text{sphere}}$.

is compared to the change in h due to the change in $\mathbf{X}_t$, $|h(\mathbf{X}_{t+1},\gamma_t,\omega_t)-h(\mathbf{X}_t,\gamma_t,\omega_t)|$. ω_t is also increased when (ii) the change in the value of the constraint function is not large enough (second inequality in Line 16). To prevent an unnecessary ill-conditioning of the problem, ω_t is decreased whenever conditions (i) and (ii) are not satisfied.

4 Empirical Results

We evaluate Algorithm 1 on the sphere function (f_{sphere}), two ellipsoid functions (f_{elli}) with condition numbers $\alpha = 10^2, 10^6$, f^2_{sphere}, $f^{0.5}_{\text{sphere}}$, the different powers function ($f_{\text{diff\_pow}}$), and the Rosenbrock function (f_{rosen}), with one linear inequality constraint. The functions are defined in Table 1. We consider the case where the constraint is active at the optimum $\mathbf{x}_{\text{opt}}$, i.e. $g(\mathbf{x}_{\text{opt}}) = 0$. We choose the optimum to be at $\mathbf{x}_{\text{opt}} = (10,\cdots,10)^\mathsf{T}$ and construct the constraint function, $g(\mathbf{x}) = \mathrm{b}^\mathsf{T}\mathbf{x}+c$, so that the KKT stationarity condition is satisfied at $\mathbf{x}_{\text{opt}}$ with $\gamma_{\text{opt}} = 1$. Therefore,

$$\mathrm{b} = -\nabla_{\mathbf{x}} f_{\cdot}(\mathbf{x}_{\text{opt}})^\mathsf{T} \quad \text{and} \quad c = \nabla_{\mathbf{x}} f_{\cdot}(\mathbf{x}_{\text{opt}})\mathbf{x}_{\text{opt}} \ ,$$

for each function. Note that all considered functions are differentiable at $\mathbf{x}_{\text{opt}} = (10,\cdots,10)^\mathsf{T}$.

For the step-size and the covariance matrix adaptation, we use the Python implementation of CMA-ES whose source code can be found at [1], with the default parameter setting detailed in [6]. We run the algorithm 11 times in $n = 10$, with $\mathbf{X}_0$ sampled uniformly in $[-5,5]^n$, $\sigma_0 = 1$, $\gamma_0 = 5$, and $\omega_0 = 1$. The results are presented for one run in Figures 1 (f_{sphere}, f^2_{sphere}, and $f^{0.5}_{\text{sphere}}$) and 2 (f_{elli} with $\alpha = 10^2, 10^6$, $f_{\text{diff\_pow}}$, and f_{rosen}). On the left column of each figure are graphs of the evolution of the distance to the optimum $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$, the step-size σ_t, the distance to the Lagrange multiplier $|\gamma_t - \gamma_{\text{opt}}|$, and the penalty factor ω_t in log-scale. On the right column of the figures are graphs representing the evolution of the coordinates of the mean vector $\mathbf{X}_t$.

Graphs on the right column of Figures 1 and 2 show the overall convergence of the algorithm to $\mathbf{x}_{\text{opt}}$. We also observe linear convergence of $\mathbf{X}_t$ to $\mathbf{x}_{\text{opt}}$, as well as linear convergence of γ_t to γ_{opt} and σ_t to 0 (left column of Figures 1 and 2). Moreover, $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$, $|\gamma_t - \gamma_{\text{opt}}|$, and σ_t decrease at the same rate. On the other hand, the penalty factor ω_t is observed to converge to a stationary value after a certain number of iterations. We sometimes observe a stagnation in graphs of $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$ due to numerical precision.

The largest convergence rate (when excluding the initial adaptation phase) is observed on f_{sphere} and the smallest one on $f^{0.5}_{\text{sphere}}$, where there is a factor of approximately 1.5 between

Algorithm 1 $(\mu/\mu_W,\lambda)$-MSR-CMA-ES with Augmented Lagrangian Constraint Handling

0 **given** $n \in \mathbb{N}_>$, $\chi = 2^{1/n}$, $k_1 = 3$, $k_2 = 5$, $\mu, \lambda \in \mathbb{N}_>$, $j = 0.3\lambda$, $0 \leq w_i < 1$, $\sum_{i=1}^{\mu} w_i = 1$,

$$\mu_{\text{eff}} = 1/\sum_{i=1}^{\mu} w_i^2,\ c_\sigma = 0.3,\ d_\sigma = 2 - 2/n,\ c_c = \frac{4+\mu_{\text{eff}}/n}{n+4+2\mu_{\text{eff}}/n}$$

$$c_1 = \frac{2}{(n+1.3)^2+\mu_{\text{eff}}},\ c_\mu = \min\left(1-c_1, 2\frac{\mu_{\text{eff}}-2+1/\mu_{\text{eff}}}{(n+2)^2+\mu_{\text{eff}}}\right)$$

1 **initialize** $\mathbf{X}_0 \in \mathbb{R}^n$, $\sigma_0 \in \mathbb{R}^+_>$, $\mathbf{C}_0 = \mathbf{I}_{n\times n}$, $t = 0$, $q_0 = 0$, $\boldsymbol{p}_0 = \mathbf{0}$, constrained_problem

2 **if** constrained_problem // **true** if the problem is constrained, **false** otherwise

3 **initialize** $\gamma_0 \in \mathbb{R}$, $\omega_0 \in \mathbb{R}^+_>$

4 **while** stopping criteria not met

5 $\mathbf{X}^i_{t+1} = \mathbf{X}_t + \sigma_t \mathbf{U}^i_{t+1}$, $\mathbf{U}^i_{t+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_t), i = 1, \cdots, \lambda$ // sample candidate solutions

6 Extract indices $\{1:\lambda, \cdots, \lambda:\lambda\}$ of ordered candidate solutions such that

$$\begin{cases} h(\mathbf{X}^{1:\lambda}_{t+1}, \gamma_t, \omega_t) \leq \ldots \leq h(\mathbf{X}^{\lambda:\lambda}_{t+1}, \gamma_t, \omega_t) & \text{if constrained\_problem} \\ f(\mathbf{X}^{1:\lambda}_{t+1}) \leq \ldots \leq f(\mathbf{X}^{\lambda:\lambda}_{t+1}) & \text{otherwise} \end{cases}$$

7 $\mathbf{X}_{t+1} = \sum_{i=1}^{\mu} w_i \mathbf{X}^{i:\lambda}_{t+1} = \mathbf{X}_t + \sigma_t \sum_{i=1}^{\mu} w_i \mathbf{U}^{i:\lambda}_{t+1}$ // recombine μ best candidate solutions

8 $K_{\text{succ}} = \begin{cases} \sum_{i=1}^{\lambda} \mathbf{1}_{\{h(\mathbf{X}^i_{t+1}, \gamma_t, \omega_t) \leq h(\mathbf{X}^{j:\lambda}_t, \gamma_t, \omega_t)\}} & \text{if constrained\_problem} \\ \sum_{i=1}^{\lambda} \mathbf{1}_{\{f(\mathbf{X}^i_{t+1}) \leq f(\mathbf{X}^{j:\lambda}_t)\}} & \text{otherwise} \end{cases}$

9 $z_t = \frac{2}{\lambda}\left(K_{\text{succ}} - \frac{\lambda}{2}\right)$ // compute success measure

10 $q_{t+1} = (1-c_\sigma)q_t + c_\sigma z_t$

11 $\sigma_{t+1} = \sigma_t \exp\left(\frac{q_{t+1}}{d_\sigma}\right)$ // update step-size

12 $\boldsymbol{p}_{t+1} = (1-c_c)\boldsymbol{p}_t + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\left(\frac{\mathbf{X}_{t+1}-\mathbf{X}_t}{\sigma_t}\right)$ // cumulation path for CMA

13 $\mathbf{C}_{t+1} = (1-c_1-c_\mu)\mathbf{C}_t + c_1 \boldsymbol{p}_{t+1}\boldsymbol{p}^\mathsf{T}_{t+1} + c_\mu \sum_{i=1}^{\mu} w_i \left(\frac{\mathbf{X}^{i:\lambda}_{t+1}-\mathbf{X}_t}{\sigma_t}\right)\left(\frac{\mathbf{X}^{i:\lambda}_{t+1}-\mathbf{X}_t}{\sigma_t}\right)^\mathsf{T}$
// update covariance matrix

14 **if** constrained_problem

15 $\gamma_{t+1} = \max(0, \gamma_t + \omega_t g(\mathbf{X}_{t+1}))$ // update Lagrange factor

16 $\omega_{t+1} = \begin{cases} \omega_t \chi^{1/4} & \text{if } \omega_t g^2(\mathbf{X}_{t+1}) < k_1 \frac{|h(\mathbf{X}_{t+1}, \gamma_t, \omega_t) - h(\mathbf{X}_t, \gamma_t, \omega_t)|}{n} \\ & \text{or } k_2 |g(\mathbf{X}_{t+1}) - g(\mathbf{X}_t)| < |g(\mathbf{X}_t)| \\ \omega_t \chi^{-1} & \text{otherwise} \end{cases}$ // update penalty factor

17 $\mathbf{X}^{j:\lambda}_{t+1} = \mathbf{X}_t + \sigma_t \mathbf{U}^{j:\lambda}_{t+1}$ // update jth best solution

18 $t = t+1$

the two convergence rates. However, there is some variance in the empirical convergence rate. In particular, on 11 performed runs we observe the highest variance in the empirical convergence rate for f_{elli} with $\alpha = 10^6$, $f_{\text{diff\_pow}}$, and f_{rosen}.

On f_{elli} with $\alpha = 10^6$, $f_{\text{diff\_pow}}$, and f_{rosen}, we observe a stagnation of $\mathbf{X}_t$ in the early stages of the algorithm (left column in Figure 2). The reason is that the adaptation of the covariance matrix takes longer on ill-conditioned problems. This explains the slow convergence of some coordinates of $\mathbf{X}_t$ to 10 (right column in Figure 2). Once the covariance matrix is adapted, the convergence occurs.

When comparing 11 single runs of Algorithm 1 to the $(1+1)$-ESs with augmented Lagrangian in [3, 4] (not shown for space reasons) on constrained f_{sphere}, f_{elli} (in $n = 10$), it appears that on f_{sphere}, Algorithm 1 needs approximately up to 1.5 times more function evaluations than algorithms in [3, 4] to reach a distance to the optimum of 10^{-4}. On f_{elli} with $\alpha = 10^2$, however, Algorithm 1 is faster and needs approximately 1.3 times less function evaluations to reach the same distance, with $\alpha = 10^6$, Algorithm 1 is around 167 times faster to reach a target of 15 (this large difference is due to the adaptation of the covariance matrix).

5 Discussion

Linear convergence is a key aspect of ESs in both unconstrained and constrained optimization scenarios. As stated in [3], the minimum requirement for a constraint handling ES is to converge linearly on convex quadratic functions with a single linear constraint. On the other hand, an algorithm for constrained optimization should be able to tackle ill-conditioned problems. Having that in mind, we proposed a $(\mu/\mu_W, \lambda)$-CMA-ES with an augmented Lagrangian approach for handling one inequality constraint, where the choice of the augmented Lagrangian constraint handling was motivated by the promising results of its implementation for the $(1+1)$-ESs with $1/5$th success rule in [3, 4]. Moreover, we showed that our algorithm—as well as $(1+1)$-ESs with augmented Lagrangian constraint handling in [3, 4]—is an instance of a more general framework for building an adaptive constraint handling algorithm from a general adaptive algorithm for unconstrained optimization.

Experiments on linearly constrained convex quadratic functions, as well as ill-conditioned functions (including the ellipsoid and Rosenbrock functions), showed linear convergence of our algorithm to the unique optimum of the constrained problem.

Acknowledgments

This work was supported by the grant ANR-2012-MONU-0009 (NumBBO) of the French National Research Agency.

References

[1] https://pypi.python.org/pypi/cma. Python source code of CMA-ES.

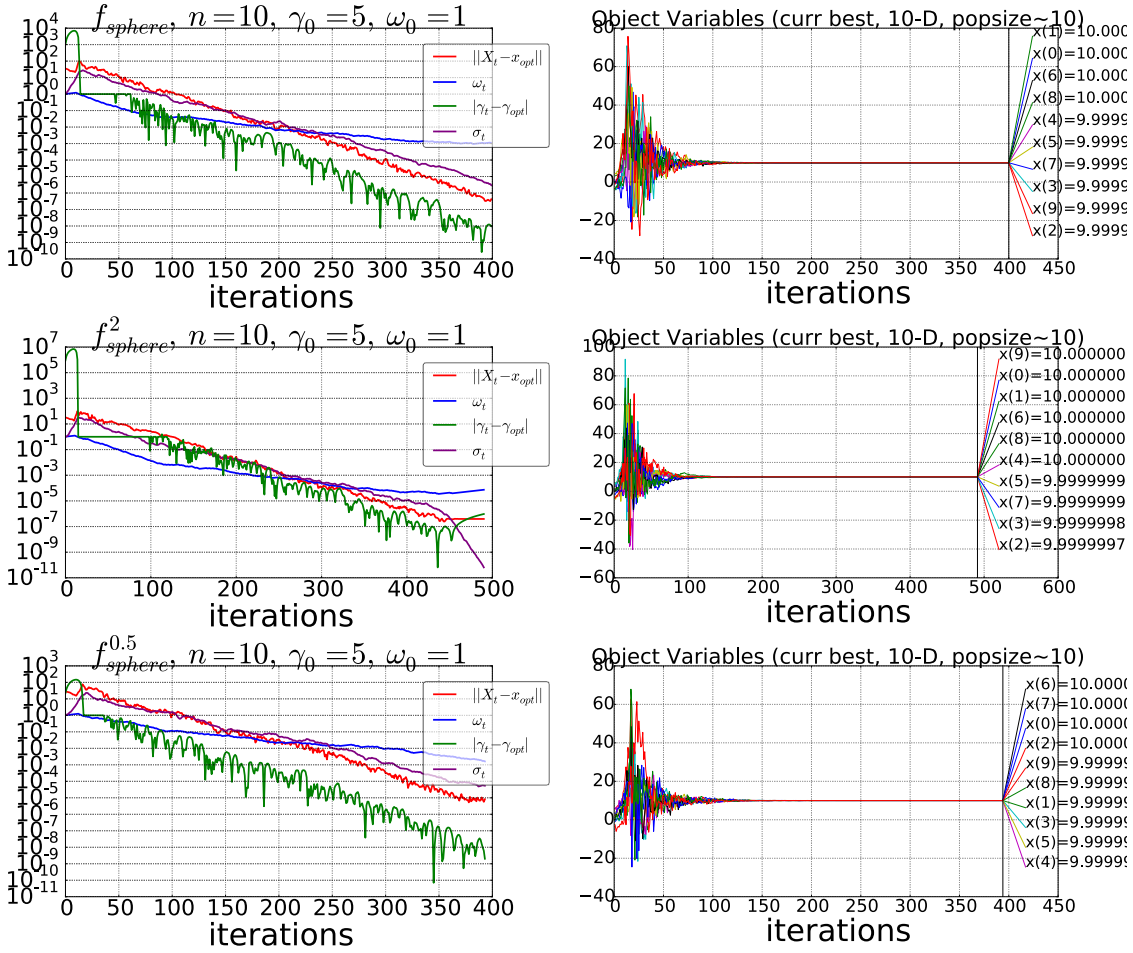

Figure 1: Single runs of $(\mu/\mu_W,\lambda)$-MSR-CMA-ES with augmented Lagrangian on f_{sphere} (top row), f^2_{sphere} (middle row), and $f^{0.5}_{\text{sphere}}$ (bottom row) in $n=10$. The optimum $\mathbf{x}_{\text{opt}}=(10,\cdots,10)^\mathsf{T}$. Left: evolution of the distance to the optimum, the distance to the Lagrange multiplier, the penalty factor, and the step-size in log-scale. Right: evolution of the coordinates of $\mathbf{X}_t$.

[2] O. Ait Elhara, A. Auger, and N. Hansen. A median success rule for non-elitist evolution strategies: Study of feasibility. In *Genetic and Evolutionary Computation Conference*, pages 415–422. ACM Press, 2013.

[3] D. V. Arnold and J. Porter. Towards au Augmented Lagrangian Constraint Handling Approach for the $(1+1)$-ES. In *Genetic and Evolutionary Computation Conference*, pages 249–256. ACM Press, 2015.

[4] A. Atamna, A. Auger, and N. Hansen. Analysis of Linear Convergence of a $(1+1)$-ES with Augmented Lagrangian Constraint Handling. In *Genetic and Evolutionary Computation Conference*, pages 213–220. ACM Press, 2016.

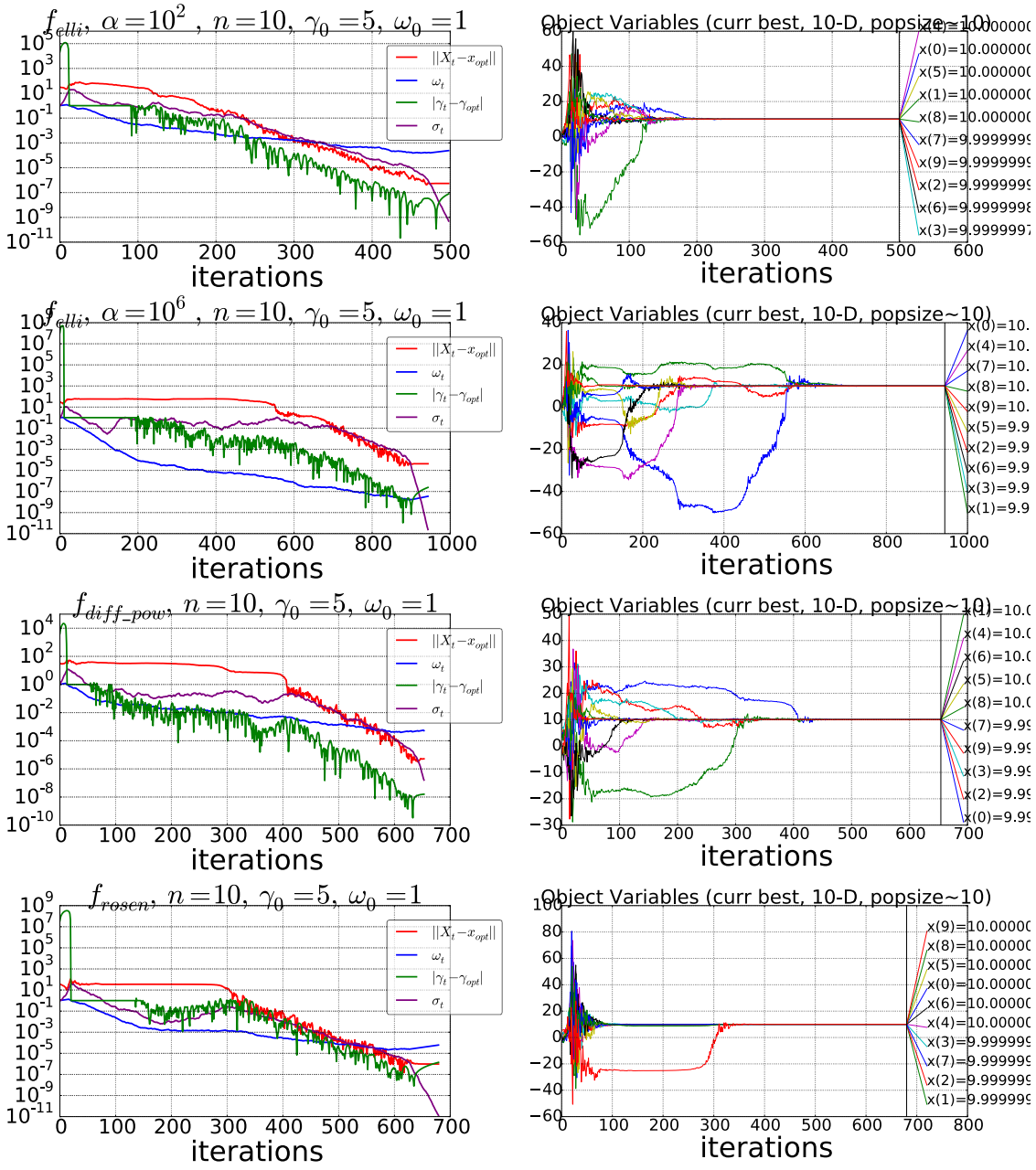

Figure 2: Single runs of $(\mu/\mu_W,\lambda)$-MSR-CMA-ES with augmented Lagrangian on f_{elli} with $\alpha=10^2$ (first row), f_{elli} with $\alpha=10^6$ (second row), $f_{\text{diff\_pow}}$ (third row), and f_{rosen} (fourth row) in $n=10$. The optimum $\mathbf{x}_{\text{opt}}=(10,\cdots,10)^\mathsf{T}$. Left: evolution of the distance to the optimum, the distance to the Lagrange multiplier, the penalty factor, and the step-size in log-scale. Right: evolution of the coordinates of $\mathbf{X}_t$.

[5] A. Auger, N. Hansen, J. Perez Zerpa, R. Ros, and M. Schoenauer. Experimental Comparisons of Derivative Free Optimization Algorithms. In Jan Vahrenhold, editor, *8th International Symposium on Experimental Algorithms*, volume 5526, pages

3–15. Springer, 2009.

[6] N. Hansen. The CMA Evolution Strategy: A Tutorial. http://arxiv.org/pdf/1604.00772v1.pdf, 2016.

[7] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[8] 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*, 11:5755–5769, 2011.

[9] M. R. Hestenes. Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[10] S. Kern, S. D. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning Probability Distributions in Continuous Evolutionary Algorithms - A Comparative Review. *Natural Computing*, 3(1):77–112, 2004.

[11] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[12] M. J. D. Powell. A Method for Nonlinear Constraints in Minimization Problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, 1969.

6.3 Linearly Convergent Evolution Strategies via Augmented Lagrangian Constraint Handling

The following paper is our latest contribution. An earlier version was submitted recently to the Foundations of Genetic Algorithms workshop. It extends the analysis we conducted in [7] with one linear constraint to the case of multiple linear constraints.

Considering a constrained optimization problem with multiple active linear inequality constraints, we present a practical adaptive augmented Lagrangian constraint handling approach. In particular, we generalize the update of the penalty parameter presented in [5] for one constraint to the case of multiple constraints. Then, using a Markov chain approach, we analyze linear convergence of a $(\mu/\mu_W, \lambda)$-ES with a general step-size adaptation rule on a modified version of the proposed practical augmented Lagrangian. By doing so, we focus the analysis on the case where all the constraints are active and manage to construct a homogeneous Markov chain which, if stable, allows to deduce linear convergence.

Linearly Convergent Evolution Strategies via Augmented Lagrangian Constraint Handling

Asma Atamna, Anne Auger, Nikolaus Hansen

Inria*
Centre Saclay–Île-de-France
LRI, Université Paris-Saclay

Abstract

We analyze linear convergence of an evolution strategy for constrained optimization with an augmented Lagrangian constraint handling approach. We study the case of multiple active linear constraints and use a Markov chain approach—used to analyze randomized optimization algorithms in the unconstrained case—to establish linear convergence under sufficient conditions. More specifically, we exhibit a class of functions on which a homogeneous Markov chain (defined from the state variables of the algorithm) exists and whose stability implies linear convergence. This class of functions is defined such that the augmented Lagrangian, centered in its value at the optimum and the associated Lagrange multipliers, is positive homogeneous of degree 2, and includes convex quadratic functions. Simulations of the Markov chain are conducted on linearly constrained sphere and ellipsoid functions to validate numerically the stability of the constructed Markov chain.

1 Introduction

Randomized (or stochastic) optimization algorithms are robust methods widely used in industry for solving continuous real-world problems. Among them, the covariance matrix adaptation (CMA) evolution strategy (ES) [12] is nowadays recognized as the state-of-the art method. It exhibits linear convergence on wide classes of functions when solving unconstrained optimization problems. However, many practical problems come with constraints and the question of how to handle them properly to particularly preserve the linear convergence is an important one [2]. Recently, an augmented Lagrangian approach to handle constraints within ES algorithms was proposed with the motivation to design an algorithm converging linearly [2]. The algorithm was analyzed theoretically and sufficient conditions

*lastname@lri.fr

for linear convergence, posed in terms of stability conditions of an underlying Markov chain, were formulated [3]. In those works, however, only the case of a single linear constraint was considered.

Markov chain theory [14] provides useful tools to analyze the linear convergence of adaptive randomized optimization algorithms and particularly evolution strategies. In a nutshell, for the case of unconstrained optimization, on scaling-invariant functions—a class of functions that includes all convex-quadratic functions—for adaptive ESs satisfying certain invariance properties (typically translation and scale-invariance), the stability analysis of an appropriate Markov chain can lead to linear convergence proofs of the original algorithm [7]. This general approach was exploited in [5] to prove the linear convergence of the $(1,\lambda)$-ES with self-adaptation on the sphere function and in [6] to prove the linear convergence of the $(1+1)$-ES with $1/5$th success rule. This general methodology to prove linear convergence in the case of unconstrained optimization was generalized to constrained optimization, in the case where a single constraint is handled via an adaptive augmented Lagrangian approach [3]. The underlying algorithm being a $(1+1)$-ES.

In this work, we generalize the study in [3] to the case of multiple linear inequality constraints. We analyze a $(\mu/\mu_W,\lambda)$-ES with an augmented Lagrangian constraint handling approach in the case of active constraints. The analyzed algorithm is an extension of the one analyzed in [3], where we generalize the original update rule for the penalty factor in [2] to the case of multiple constraints. We construct a homogeneous Markov chain for problems such that the corresponding augmented Lagrangian, centered at the optimum of the problem and the corresponding Lagrange multipliers, is positive homogeneous of degree 2, given some invariance properties are satisfied by the algorithm. Then, we give sufficient stability conditions on the Markov chain such that the algorithm converges to the optimum of the constrained problem as well as to the associated Lagrange multipliers. Finally, the stability of the constructed Markov chain is investigated empirically.

The rest of this paper is organized as it follows: we present augmented Lagrangian methods in Section 2 and give an overview on how the Markov chain approach is used to prove linear convergence in the unconstrained case in Section 3. We formally define the studied optimization problem, as well as the considered augmented Lagrangian in Sections 4 and 5 respectively. In Section 6, we present the studied algorithm and discuss its invariance properties. In Section 7, we present the constructed Markov chain and deduce linear convergence given its stability. Finally, we present our empirical results in Section 8 and conclude with a discussion in Section 9.

Notations

The notations that are not explicitly defined in the paper are presented here. We denote $\mathbb{R}^+$ the set of positive real numbers, $\mathbb{R}^+_>$ the set of strictly positive real numbers, and $\mathbb{N}_>$ the set of natural numbers without 0. $\mathbf{x} \in \mathbb{R}^n$ is a column vector, $\mathbf{x}^\intercal$ is its transpose, and $\mathbf{0} \in \mathbb{R}^n$ is the zero vector. $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$, $[\mathbf{x}]_i$ its ith element, and $[\mathbf{M}]_{ij}$ the element in the ith row and jth column of matrix $\mathbf{M}$. $\mathbf{I}_{n\times n} \in R^{n\times n}$ denotes the identity matrix, $\mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})$ the multivariate standard normal distribution, and $\sim$ the equality in

distribution. The symbol $\circ$ is the function composition operator. The derivative with respect to $\mathbf{x}$ is denoted $\nabla_{\mathbf{x}}$ and the expectation of a random variable $X \sim \pi$ is denoted E_π.

2 Augmented Lagrangian Methods

Augmented Lagrangian methods are constraint handling approaches that combine penalty function methods with the Karush-Kuhn-Tucker (KKT) necessary conditions of optimality. They were first introduced in [13, 16] to overcome the limitations of penalty function methods—in particular quadratic penalty methods—which suffer from ill-conditioning as the penalty parameters need to tend to infinity in order to converge [15].

Similarly to penalty methods, augmented Lagrangian methods transform the constrained problem into one or more unconstrained problems where an augmented Lagrangian, consisting in a Lagrangian part and a penalty function part, is optimized. The Lagrangian is a function $\mathscr{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ defined as

$$\mathscr{L}(\mathbf{x}, \gamma) = f(\mathbf{x}) + \sum_{i=1}^{m} \gamma^i g_i(\mathbf{x}) \ , \tag{1}$$

for a function f subject to m constraints $g_i(\mathbf{x}) \leq 0$. The vector $\gamma = (\gamma^1, \cdots, \gamma^m)^\intercal \in \mathbb{R}^m$ represents the Lagrange factors. An important property of $\mathscr{L}$ is the so-called KKT stationarity condition which states that, given some regularity conditions (constraint qualifications) are satisfied, if $\mathbf{x}^* \in \mathbb{R}^n$ is a local optimum of the constrained problem, then there exists a vector $\gamma^* = (\gamma^{*1}, \cdots, \gamma^{*m})^\intercal \in (\mathbb{R}^+)^m$ of Lagrange multipliers γ^{*i}, $i = 1, \cdots, m$, such that

$$\nabla_{\mathbf{x}} \mathscr{L}(\mathbf{x}^*, \gamma^*) = \nabla_{\mathbf{x}} f(\mathbf{x}^*) + \sum_{i=1}^{m} \gamma^{*i} \nabla g_i(\mathbf{x}^*) = \mathbf{0} \ ,$$

if we assume f and g_i, $i = 1, \cdots, m$, are differentiable at $\mathbf{x}^*$.

Remark 1. Given the gradients $\nabla_{\mathbf{x}} f(\mathbf{x}^*)$ and $\nabla_{\mathbf{x}} g_i(\mathbf{x}^*)$, $i = 1, \cdots, m$, exist, the first-order necessary conditions of optimality (KKT conditions) ensure the existence of at least one vector γ^* of Lagrange multipliers. However, if the constraints satisfy the linear independence constraint qualification (LICQ), that is, the set of constraint normals is linearly independent, the vector γ^* of Lagrange multipliers is unique [15].

The Lagrangian $\mathscr{L}$ is combined to a penalty function, which is a function of the constraints g_i, to construct the augmented Lagrangian h. Examples of augmented Lagrangians are given in (9) and (10), where $\omega = (\omega^1, \cdots, \omega^m)^\intercal \in (\mathbb{R}^+_>)^m$ is the vector of the penalty factors ω^i. More generally, the augmented Lagrangian can be defined as

$$h(\mathbf{x}, \gamma, \omega) = f(\mathbf{x}) + \sum_{i=1}^{m} \varphi(g_i(\mathbf{x}), \gamma^i, \omega^i) \ , \tag{2}$$

where φ is chosen such that a local optimum $\mathbf{x}^*$ of the constrained problem is a stationary point of h, that is for all $\gamma \in (\mathbb{R}^+_>)^m$,

$$\nabla_{\mathbf{x}} h(\mathbf{x}^*, \gamma^*, \omega) = \mathbf{0} \ ,$$

assuming the gradients at $\mathbf{x}^*$ are defined. The augmented Lagrangian h is minimized for given values of γ and ω instead of the initial objective function f.

In adaptive augmented Lagrangian approaches, γ is adapted to approach the Lagrange multipliers and ω is adapted to guide the search towards feasible solutions. A proper adaptation mechanism for ω helps preventing ill-conditioning since, with an augmented Lagrangian approach, the penalty factors ω^i do not need to tend to infinity to achieve convergence [15].

There exist in the literature some examples where augmented Lagrangian approaches are used in the context of evolutionary algorithms. In [17], the authors present a coevolutionary method for constrained optimization with an augmented Lagrangian approach, where two populations (one for the parameter vector and one for Lagrange factors) are evolved in parallel, using an evolution strategy with self-adaptation. The approach is tested on four non-linear constrained problems, with a fixed value for the penalty parameter.

In [9], the authors present an augmented-Lagrangian-based genetic algorithm for constrained optimization. Their algorithm requires a local search procedure for improving the current best solution in order to converge to the optimal solution and to the associated Lagrange multipliers.

More recently, an augmented Lagrangian approach was combined with a $(1+1)$-ES for the case of a single linear constraint [2]. An update rule was presented for the penalty parameter and the algorithm was observed to converge on the sphere function and on a moderately ill-condition ellipsoid function, with one linear constraint. This algorithm was analyzed in [3] using tools from the Markov chain theory. The authors constructed a homogeneous Markov chain and deduced linear convergence under the stability of this Markov chain. In [4], the augmented Lagrangian constraint handling mechanism in [2] is implemented for CMA-ES and a general framework for building a general augmented Lagrangian based randomized algorithm for constrained optimization in the case of one constraint is presented.

3 Markov Chain Analysis and Linear Convergence

Randomized or stochastic optimization algorithms are iterative methods where—most often—the state of the algorithm is a Markov chain. For a certain class of algorithms obeying proper invariance properties, Markov chain theory can provide powerful tools to prove the linear convergence of the algorithms [8, 7, 5]. We illustrate here on a simple case the general methodology to prove linear convergence of an adaptive randomized algorithm using Markov chain theory. We assume for the sake of simplicity the minimization of the sphere function $\mathbf{x} \mapsto f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\intercal\mathbf{x}$ with, without loss of generality, the optimum in zero. We assume that the state of the algorithm at iteration t is given by the current estimate $\mathbf{X}_t$ of the optimum and a positive factor, the step-size σ_t. From this state, λ new candidate solutions are sampled according to

$$\mathbf{X}_{t+1}^i = \mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^i,\ i = 1,\ldots,\lambda \quad ,$$

where $\mathbf{U}_{t+1}^i$ are independent identically distributed (i.i.d.) standard multivariate normal distributions (with mean zero and covariance matrix identity). The state of the algorithm is

then updated via two deterministic update functions $\mathscr{G}_{\mathbf{x}}$ and $\mathscr{G}_{\sigma}$ according to

$$\mathbf{X}_{t+1} = \mathscr{G}_{\mathbf{x}}((\mathbf{X}_t, \sigma_t), \varsigma * \mathbf{U}_{t+1}) \ , \tag{3}$$
$$\sigma_{t+1} = \mathscr{G}_{\sigma}(\sigma_t, \varsigma * \mathbf{U}_{t+1}) \ , \tag{4}$$

where $\mathbf{U}_{t+1} = [\mathbf{U}_{t+1}^1, \cdots, \mathbf{U}_{t+1}^{\lambda}]$ is the vector of i.i.d. random vectors $\mathbf{U}_{t+1}^i$ and

$$\varsigma = Ord(f(\mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^i)_{i=1,\cdots,\lambda})$$

is the permutation that contains the indices of the candidate solutions $\mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^i$ ranked according to their f-value. That is, the ordering is done using the operator Ord such that, given λ real numbers $z_1, \cdots, z_\lambda$, $\varsigma = Ord(z_1, \cdots, z_\lambda)$ satisfies

$$z_{\varsigma(1)} \leq \cdots \leq z_{\varsigma(\lambda)} \ . \tag{5}$$

In (3) and (4), the operator '*' applies the permutation ς to $\mathbf{U}_{t+1}$ and

$$\varsigma * \mathbf{U}_{t+1} = [\mathbf{U}_{t+1}^{\varsigma(1)}, \cdots, \mathbf{U}_{t+1}^{\varsigma(\lambda)}] \ . \tag{6}$$

It has been shown that if the update functions $\mathscr{G}_{\mathbf{x}}$ and $\mathscr{G}_{\sigma}$ satisfy the following conditions [7]:

(i) for all $\mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^n$, for all $\sigma > 0$, for all $\mathbf{y} \in (\mathbb{R}^n)^{\lambda}$

$$\mathscr{G}_{\mathbf{x}}((\mathbf{x} + \mathbf{x}_0, \sigma), \mathbf{y}) = \mathscr{G}_{\mathbf{x}}((\mathbf{x}, \sigma), \mathbf{y}) + \mathbf{x}_0 \ ,$$

(ii) for all $\mathbf{x} \in \mathbb{R}^n$, for all $\alpha, \sigma > 0$, for all $\mathbf{y} \in (\mathbb{R}^n)^{\lambda}$

$$\mathscr{G}_{\mathbf{x}}((\mathbf{x}, \sigma), \mathbf{y}) = \alpha \mathscr{G}_{\mathbf{x}}\left(\left(\frac{\mathbf{x}}{\alpha}, \frac{\sigma}{\alpha}\right), \mathbf{y}\right) \ ,$$

(iii) for all $\alpha, \sigma > 0$, for all $\mathbf{y} \in (\mathbb{R}^n)^{\lambda}$

$$\mathscr{G}_{\sigma}(\sigma, \mathbf{y}) = \alpha \mathscr{G}_{\sigma}\left(\frac{\sigma}{\alpha}, \mathbf{y}\right) \ ,$$

then the algorithm is translation-invariant and scale-invariant. As a consequence, $(\mathbf{Y}_t)_{t \in \mathbb{N}}$, with $\mathbf{Y}_t = \frac{\mathbf{X}_t}{\sigma_t}$, is a homogeneous Markov chain that can be defined independently of $(\mathbf{X}_t, \sigma_t)$, given $\mathbf{Y}_0 = \frac{\mathbf{X}_0}{\sigma_0}$, as

$$\mathbf{Y}_{t+1} = \frac{\mathscr{G}_{\mathbf{x}}((\mathbf{Y}_t, 1), \varsigma * \mathbf{U}_{t+1})}{\mathscr{G}_{\sigma}(1, \varsigma * \mathbf{U}_{t+1})} \ ,$$

where $\varsigma = Ord(f(\mathbf{Y}_t + \mathbf{U}_{t+1}^i)_{i=1,\cdots,\lambda})$ [7, Proposition 4.1] (this result is true for the sphere function but more generally for a scaling-invariant objective function). Let consider now the following definition of linear convergence:

Definition 1. We say that a sequence $(\mathbf{X}_t)_{t\in\mathbb{N}}$ of random vectors $\mathbf{X}_t$ converges linearly almost surely (a.s.) to $\mathbf{x}_{\text{opt}}$ if there exists $\text{CR} > 0$ such that

$$\lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|} = -\text{CR} \text{ a.s.}$$

The constant CR is called the convergence rate.

Using the property of the logarithm, the quantity $\frac{1}{t} \ln \frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|}$ ($\mathbf{x}_{\text{opt}} = \mathbf{0}$ here) can be expressed as a function of $\mathbf{Y}_t$ according to

$$\begin{aligned}\frac{1}{t} \ln \frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|} = \frac{1}{t} \sum_{k=0}^{t-1} \ln \frac{\|\mathbf{X}_{k+1}\|}{\|\mathbf{X}_k\|} &= \frac{1}{t} \sum_{k=0}^{t-1} \ln \frac{\|\mathbf{X}_{k+1}\|}{\|\mathbf{X}_k\|} \frac{\sigma_k \mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1})}{\sigma_{k+1}} \\ &= \frac{1}{t} \sum_{k=0}^{t-1} \ln \frac{\|\mathbf{Y}_{k+1}\|}{\|\mathbf{Y}_k\|} \mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1}) \ , \end{aligned} \tag{7}$$

where we have successively artificially introduced $\sigma_{k+1} = \sigma_k \mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1})$ and then used that $\mathbf{Y}_k = \mathbf{X}_k/\sigma_k$ and $\mathbf{Y}_{k+1} = \mathbf{X}_{k+1}/\sigma_{k+1}$. In (7), we have expressed the term whose limit we are interested in as the empirical average of a function of a Markov chain. However, we know from Markov chain theory that if some sufficient stability conditions—given for instance in Theorem 17.0.1 from [14]—are satisfied by $(\mathbf{Y}_t)_{t\in\mathbb{N}}$, then a law of large numbers (LLN) for Markov chains can be applied to the right-hand side of the previous equation. Consequently,

$$\begin{aligned}\lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|} &= \lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t-1} \ln \frac{\|\mathbf{Y}_{k+1}\|}{\|\mathbf{Y}_k\|} \mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1}) \\ &= \int \ln \|\mathbf{y}\| \pi(d\mathbf{y}) - \int \ln \|\mathbf{y}\| \pi(d\mathbf{y}) + \underbrace{\int E(\ln(\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{t+1})) | \mathbf{Y}_t = \mathbf{y}) \pi(d\mathbf{y})}_{-\text{CR}} \ , \end{aligned}$$

where π is the invariant probability measure of the Markov chain $(\mathbf{Y}_t)_{t\in\mathbb{N}}$. Hence, assuming that a law of large number holds for the Markov chain $(\mathbf{Y}_t)_{t\in\mathbb{N}}$, the algorithm described by the iterative sequence $(\mathbf{X}_t, \sigma_t)_{t\in\mathbb{N}}$ will converge linearly at the rate expressed as minus the expected log step-size change (where the expectation is taken with respect to the invariant probability measure of $(\mathbf{Y}_t)_{t\in\mathbb{N}}$). This methodology to prove the linear convergence of adaptive algorithms (including many evolution strategies) in the unconstrained case holds on scaling-invariant functions (that include particularly functions that write $g \circ f$, where g is a 1-D strictly increasing function and f is positively homogeneous, typically f can be a convex-quadratic function). It provides the *exact* expression of the convergence rate that equals the expected log step-size change with respect to the stationary distribution of a Markov chain. This illustrates that Markov chains are central tools for the analysis of convergence of adaptive randomized optimization algorithms. Remark that the convergence rate can be easily simulated to obtain quantitative estimates and dependencies with respect to internal parameters of the algorithm or of the objective functions.

We see that there are two distinct steps for the analysis of the linear convergence:

(i) Identify on which class of functions the algorithms we study can exhibit a Markov chain whose stability will lead to the linear convergence of the underlying algorithm (in the example above, the Markov chain equals $\mathbf{Y}_t = \mathbf{X}_t/\sigma_t$).

(ii) Prove the stability of the identified Markov chain.

The second step is arguably the most complex one. So far, it has been successfully achieved for the analysis of the linear convergence of self-adaptive evolution strategies [5] and for the $(1+1)$-ES with one-fifth success rule [6] in the unconstrained case. The main tools to prove the stability rely on Foster-Lyapunov drift conditions [14]. In this paper, we will focus on the first step. Particularly, the Markov chain for step-size adaptive randomized search optimizing scaling-invariant functions (i.e. unconstrained optimization) was identified in [7]. In addition, in the constrained case, the Markov chain has been identified for the $(1+1)$-ES with an augmented Lagrangian constraint handling in the case of one linear inequality constraint [3]. We consider here the extension to more than one constraint and a more general algorithm framework.

4 Optimization Problem

We consider throughout this work the problem of minimizing a function f subject to m linear inequality constraints $g_i(\mathbf{x}) \leq 0$, $i = 1, \cdots, m$. Formally, this writes

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ &\text{subject to } g_i(\mathbf{x}) \leq 0,\ i = 1, \cdots, m \ , \end{aligned} \tag{8}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g_i : \mathbb{R}^n \to \mathbb{R}$, and $g_i(\mathbf{x}) = \mathrm{b}_i{}^{\mathsf{T}}\mathbf{x} + c_i$, $\mathrm{b}_i \in \mathbb{R}^n$, $c_i \in \mathbb{R}$.

We assume this problem to have a unique global optimum $\mathbf{x}_{\text{opt}}$. We also assume the constraints to be active at $\mathbf{x}_{\text{opt}}$, that is, $g_i(\mathbf{x}_{\text{opt}}) = 0$, $i = 1, \cdots, m$. This constitutes the most difficult case. Indeed, if the constraint is not active, when close enough to the optimum, the algorithm will typically not see the constraint such that it will behave as in the unconstrained case. In terms of theoretical analysis, the unconstrained case—for a general class of step-size adaptive algorithms—is well understood in the case of scaling-invariant functions [7]. Additionally, we assume that the gradients at $\mathbf{x}_{\text{opt}}$, $\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}})$ and $\nabla_{\mathbf{x}} g_i(\mathbf{x}_{\text{opt}})$, $i = 1, \cdots, m$, are defined and that the constraints satisfy the linear independence constraint qualification (LICQ) [15] at $\mathbf{x}_{\text{opt}}$. We denote γ_{opt} the (unique) vector of Lagrange multipliers associated to $\mathbf{x}_{\text{opt}}$.

5 Considered Augmented Lagrangian

A practical augmented Lagrangian for the optimization problem in (8) is given in the following equation

$$h(\mathbf{x},\gamma,\omega) = f(\mathbf{x}) + \sum_{i=1}^{m} \underbrace{\begin{cases} \gamma^i g_i(\mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x})^2 & \text{if } \gamma^i + \omega^i g_i(\mathbf{x}) \geq 0 \\ -\frac{{\gamma^i}^2}{2\omega^i} & \text{otherwise} \end{cases}}_{\varphi_1(g_i(\mathbf{x}),\gamma^i,\omega^i)} \quad . \tag{9}$$

The use of a different penalty factor for each constraint is motivated by the fact that the penalization should depend on the constraint violation—which might be different for different constraints. The quality of a solution $\mathbf{x}$ is evaluated by adding $f(\mathbf{x})$ and either (i) $\gamma^i g_i(\mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x})^2$ if $g_i(\mathbf{x})$ is larger than $-\frac{\gamma^i}{\omega^i}$ or (ii) $-\frac{{\gamma^i}^2}{2\omega^i}$ otherwise, for each constraint function g_i.

The augmented Lagrangian in (9) is constructed such that (i) the fitness function remains unchanged when far in the feasible domain and (ii) h is "smooth" in that it is differentiable with respect to g_i. Therefore, (9) is the recommended augmented Lagrangian in practice. For the analysis, however, we consider a simpler augmented Lagrangian (equation below) so that we can construct a Markov chain.

$$h(\mathbf{x},\gamma,\omega) = f(\mathbf{x}) + \sum_{i=1}^{m} \underbrace{\gamma^i g_i(\mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x})^2}_{\varphi_2(g_i(\mathbf{x}),\gamma^i,\omega^i)} \quad . \tag{10}$$

The difference is that in the previous formulation the penalization is a constant and hence inconsequential for $g_i(\mathbf{x}) < -\gamma^i/\omega^i$. Since we focus in our study on problems where the constraints are active at the optimum, the augmented Lagrangians in (9) and (10) are equivalent in the vicinity of $\mathbf{x}_{\text{opt}}$, as illustrated in Figure 1 for one constraint. Inactive constraints are covered in that the analysis remains valid when these constraints are removed, in which case we recover the original equation (9) up to adding a constant to the f-value. Therefore, conducting the analysis with (10) gives insight into how a practical algorithm using (9) would perform close to the optimum.

6 Algorithm

In this section, we present a general ES (Algorithm 1) with comma-selection and weighted recombination (denoted $(\mu/\mu_W,\lambda)$-ES) for constrained optimization, where the constraints are handled using an augmented Lagrangian approach.

First, λ i.i.d. vectors $\mathbf{U}_{t+1}^i$ are sampled in Line 3 of Algorithm 1 according to a normal distribution of mean $\mathbf{0}$ and covariance matrix the identity. They are used to create λ candidate solutions $\mathbf{X}_{t+1}^i$ according to

$$\mathbf{X}_{t+1}^i = \mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^i \quad , \tag{11}$$

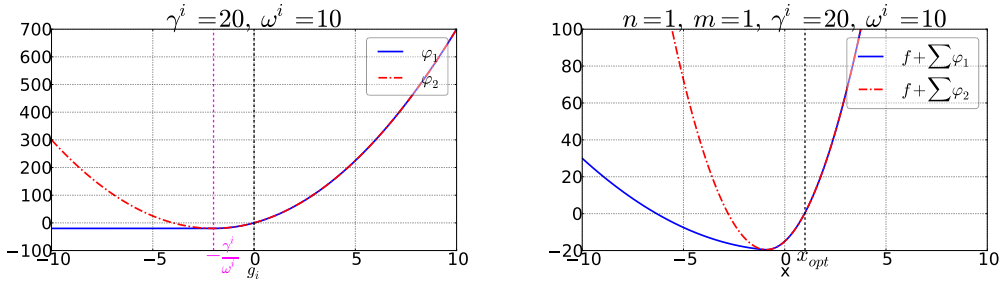

Figure 1: Left: $\varphi_j(g_i(\mathbf{x}), \gamma^i, \omega^i)$ for $j = 1$ (blue) and $j = 2$ (red), as a function of g_i. Right: Augmented Lagrangians, $f(\mathbf{x}) + \sum_{i=1}^{m} \varphi_j(g_i(\mathbf{x}), \gamma^i, \omega^i)$, for $j = 1$ (blue) and $j = 2$ (red), in $n = 1$ with $m = 1$. $f(x) = \frac{1}{2}x^2$, $g_1(x) = x - 1$, and $x_{\text{opt}} = 1$.

where $\mathbf{X}_t$ is the current estimate of the optimum and σ_t is the step-size. The candidate solutions are then ranked according to their fitness, determined by their h-value. This is done in Line 4 with the operator *Ord* defined in (5), where ς is the permutation that contains the indices of the ordered candidate solutions.

Later on, we will make explicit the dependency of ς on the objective function, the current solution, and the current step-size, where needed (this would read $\varsigma_{(\mathbf{X}_t,\sigma_t)}^{h(\cdot,\gamma_t,\omega_t)}$ here). The solution $\mathbf{X}_{t+1}$ at the next iteration is computed by recombining the μ best candidate solutions—or parents—in a weighted sum according to Line 5, where w_i, $i = 1, \cdots, \mu$, are the weights associated to the parents and the operator '*' applies the permutation ς to the vector $\mathbf{U}_{t+1}$ of the sampled vectors $\mathbf{U}_{t+1}^i$ as defined in (6).

The step-size is adapted in Line 6 using a general update function $\mathcal{G}_\sigma$. For the sake of simplicity, we consider that $\mathcal{G}_\sigma$ is a function of the current step-size σ_t and the ranked vector $\varsigma * \mathbf{U}_{t+1}$ of the sampled vectors $\mathbf{U}_{t+1}^i$.

The Lagrange factors are adapted in Line 7. As a result of this update rule, a Lagrange factor γ_t^i is increased if $g_i(\mathbf{X}_{t+1})$ is positive and decreased otherwise. A damping factor d_γ is used to attenuate the change in the value of γ_t^i.

Each penalty factor ω_t^i is adapted according to Line 8. This update is a generalization to the case of many constraints of the original update proposed in [2] for the case of a single constraint. A penalty factor ω_t^i is increased in two cases: the first one is given by the first inequality in Line 8 and corresponds to the case where (i) the change in h-value due to changes in γ_t^i and ω_t^i is smaller than the change in h-value due to the change in $\mathbf{X}_t$. Indeed

$$\omega_t^i g_i(\mathbf{X}_{t+1})^2 \approx |h(\mathbf{X}_{t+1}, \gamma_t + \Delta_i\gamma, \omega_t + \Delta_i\omega) - h(\mathbf{X}_{t+1}, \gamma_t, \omega_t)| \ ,$$

where $\Delta_i\gamma = (0, \cdots, \Delta\gamma^i, \cdots, 0)^\intercal$ and $\Delta_i\omega = (0, \cdots, \Delta\omega^i, \cdots, 0)^\intercal$. By increasing the penalization, we prevent premature stagnation [2]. The parameter ω_t^i is also increased if (ii) the change in the corresponding constraint value $|g_i(\mathbf{X}_{t+1}) - g_i(\mathbf{X}_t)|$ is significantly smaller than $|g_i(\mathbf{X}_t)|$ (second inequality in Line 8). In this case, increasing the penalization allows approaching the constraint boundary ($g_i(\mathbf{x}) = 0$) more quickly. However, increasing ω_t^i increases the ill-conditioning of the problem at hand, therefore, in all other cases, ω_t^i is decreased (second case in Line 8). Similarly to the update of the Lagrange factors, we use a damping factor d_ω to moderate the changes in ω_t^i.

Algorithm 1 $(\mu/\mu_W,\lambda)$-ES with Augmented Lagrangian Constraint Handling

0 **given** $n \in \mathbb{N}_>,\ \chi, k_1, k_2, d_\gamma, d_\omega \in \mathbb{R}^+_>, \lambda, \mu \in \mathbb{N}_>, 0 \leq w_i < 1,\ \sum_{i=1}^{\mu} w_i = 1$

1 **initialize** $\mathbf{X}_0 \in \mathbb{R}^n,\ \sigma_0 \in \mathbb{R}^+_>,\ \gamma_0 \in \mathbb{R}^m,\ \omega_0 \in (\mathbb{R}^+_>)^m,\ t = 0$

2 **while** stopping criterion not met

3 $\quad \mathbf{U}^i_{t+1} = \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n}),\ i = 1, \cdots, \lambda$

4 $\quad \varsigma = Ord(h(\mathbf{X}_t + \sigma_t \mathbf{U}^i_{t+1}, \gamma_t, \omega_t)_{i=1,\cdots,\lambda})$

5 $\quad \mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \sum_{i=1}^{\mu} w_i [\varsigma * \mathbf{U}_{t+1}]_i, \mathbf{U}_{t+1} = [\mathbf{U}^1_{t+1}, \cdots, \mathbf{U}^\lambda_{t+1}]$

6 $\quad \sigma_{t+1} = \mathcal{G}_\sigma(\sigma_t, \varsigma * \mathbf{U}_{t+1})$

7 $\quad \gamma^i_{t+1} = \gamma^i_t + \frac{\omega^i_t}{d_\gamma} g_i(\mathbf{X}_{t+1}),\ i = 1, \cdots, m$

8 $\quad \omega^i_{t+1} = \begin{cases} \omega^i_t \chi^{1/(4d_\omega)} & \text{if } \omega^i_t g_i(\mathbf{X}_{t+1})^2 < k_1 \frac{|h(\mathbf{X}_{t+1},\gamma_t,\omega_t) - h(\mathbf{X}_t,\gamma_t,\omega_t)|}{n} \\ & \text{or } k_2 |g_i(\mathbf{X}_{t+1}) - g_i(\mathbf{X}_t)| < |g_i(\mathbf{X}_t)| \\ \omega^i_t \chi^{-1/d_\omega} & \text{otherwise, } i = 1, \cdots, m \end{cases}$

9 $\quad t = t + 1$

Algorithm 1 is a randomized adaptive algorithm that can be defined in an abstract manner as follows: given the state variables $(\mathbf{X}_t, \sigma_t, \gamma_t, \omega_t)$ at iteration t, a transition function $\mathscr{F}^{(f,\{g_i\}_{i=1,\cdots,m})}$, and the vector $\mathbf{U}_{t+1} = [\mathbf{U}^1_{t+1}, \cdots, \mathbf{U}^\lambda_{t+1}]$ of i.i.d. normal vectors $\mathbf{U}^i_{t+1}$, compute the state at iteration $t+1$ according to

$$(\mathbf{X}_{t+1}, \sigma_{t+1}, \gamma_{t+1}, \omega_{t+1}) = \mathscr{F}^{(f,\{g_i\}_{i=1,\cdots,m})}((\mathbf{X}_t, \sigma_t, \gamma_t, \omega_t), \mathbf{U}_{t+1}),$$

where the superscript indicates the objective function to minimize, f, and the constraint functions, g_i. The deterministic transition function $\mathscr{F}^{(f,\{g_i\}_{i=1,\cdots,m})}$ is defined by the following general update rules for $\mathbf{X}_t$, σ_t, γ_t, and ω_t:

$$\mathbf{X}_{t+1} = \mathscr{G}_\mathbf{x}((\mathbf{X}_t, \sigma_t), \varsigma * \mathbf{U}_{t+1}) \ , \tag{12}$$

$$\sigma_{t+1} = \mathscr{G}_\sigma(\sigma_t, \varsigma * \mathbf{U}_{t+1}) \ , \tag{13}$$

$$\gamma^i_{t+1} = \mathscr{H}^{g_i}_\gamma(\gamma^i_t, \omega^i_t, \mathbf{X}_{t+1}),\ i = 1, \cdots, m \ , \tag{14}$$

$$\omega^i_{t+1} = \mathscr{H}^{(f,g_i)}_\omega(\omega^i_t, \gamma^i_t, \mathbf{X}_t, \mathbf{X}_{t+1}),\ i = 1, \cdots, m \ , \tag{15}$$

where ς, $\mathscr{G}_\mathbf{x}$, $\mathscr{H}_\gamma$, and $\mathscr{H}_\omega$ are given in Lines 4, 5, 7, and 8 of Algorithm 1 respectively. These notations are particularly useful for defining the notions of translation and scale-invariance in the next subsection. They also make the connection between the constructed homogeneous Markov chain and the original algorithm clearer.

Comparing (12), (13), (14), and (15) to (3) and (4), it is easy to see that Algorithm 1 is built by taking an adaptive algorithm for unconstrained optimization and changing its

objective function to an adaptive one—the augmented Lagrangian—where the parameters of the augmented Lagrangian are additionally adapted every iteration. This idea was already put forward in [4] for the case of a single constraint, and we generalize it here to the case of m constraints.

6.1 Invariance

Invariance with respect to transformations of the search space is a central property in randomized adaptive algorithms. In the unconstrained case, it is exploited to demonstrate linear convergence [7, 6]. In this subsection, we discuss translation-invariance and scale-invariance of Algorithm 1. We first recall the definition of a group homomorphism and introduce some notations.

Definition 2. Let $(G_1,.)$ and $(G_2,*)$ be two groups. A function $\Phi : G_1 \to G_2$ is a group homomorphism if for all $x,y \in G_1$, $\Phi(x.y) = \Phi(x) * \Phi(y)$.

We denote $\mathscr{S}(\Omega)$ the set of all bijective transformations from a set Ω to itself and $\text{Homo}((\mathbb{R}^n,+),(\mathscr{S}(\Omega),\circ))$ (respectively $\text{Homo}((\mathbb{R}^+_>,.),(\mathscr{S}(\Omega),\circ))$) the set of group homomorphisms from $(\mathbb{R}^n,+)$ (respectively from $(\mathbb{R}^+_>,.)$) to $(\mathscr{S}(\Omega),\circ)$.

Translation-invariance informally translates the non-sensitivity of an algorithm with respect to the choice of its initial point, that is the algorithm will exhibit the same behavior when optimizing $\mathbf{x} \mapsto f(\mathbf{x})$ or $\mathbf{x} \mapsto f(\mathbf{x}-\mathbf{x}_0)$ for any $\mathbf{x}_0$. More formally, an algorithm is translation-invariant if we can find a state-space transformation such that optimizing $\mathbf{x} \mapsto f(\mathbf{x})$ or $\mathbf{x} \mapsto f(\mathbf{x}-\mathbf{x}_0)$ is the same up to the state-space transformation. In the next definition, which is a generalization to the constrained case of the definition given in [7], we ask that the set of state-space transformations is given via a group homomorphism from the group acting on the function to transform the functions, that is $(\mathbb{R}^n,+)$, to the group of bijective state-space transformations. Indeed this group homomorphism naturally emerges when attempting to prove invariance. More formally, we have the following definition of translation-invariance.

Definition 3. A randomized adaptive algorithm with transition function $\mathscr{F}^{(f,\{g_i\}_{i=1,\cdots,m})} : \Omega \times \mathbb{U}^\lambda \to \Omega$, where f is the objective function to minimize and g_i are the constraint functions, is translation-invariant if there exists a group homomorphism $\Phi \in$
$\text{Homo}((\mathbb{R}^n,+),(\mathscr{S}(\Omega),\circ))$ such that for any objective function f, for any constraint g_i, for any $\mathbf{x}_0 \in \mathbb{R}^n$, for any state $\mathbf{s} \in \Omega$, and for any $\boldsymbol{u} \in \mathbb{U}^\lambda$,

$$\mathscr{F}^{\left(f(\mathbf{x}),\{g_i(\mathbf{x})\}_{i=1,\cdots,m}\right)}(\mathbf{s},\boldsymbol{u}) = \Phi(-\mathbf{x}_0)\left(\mathscr{F}^{\left(f(\mathbf{x}-\mathbf{x}_0),\{g_i(\mathbf{x}-\mathbf{x}_0)\}_{i=1,\cdots,m}\right)}(\Phi(\mathbf{x}_0)(\mathbf{s}),\boldsymbol{u})\right) \quad .$$

Similarly for scale-invariance, the set of state-space transformations comes from a group homomorphism between the group where the scaling factors acting to transform the objective functions are taken from, that is the group $(\mathbb{R}^+_>,.)$ and the group of bijective state-space transformations.

Definition 4. A randomized adaptive algorithm with transition function $\mathscr{F}^{(f,\{g_i\}_{i=1,\cdots,m})} : \Omega \times \mathbb{U}^\lambda \to \Omega$, where f is the objective function to minimize and g_i are the constraint functions, is scale-invariant if there exists a group homomorphism $\Phi \in \mathrm{Homo}((\mathbb{R}_>^+,.),(\mathscr{S}(\Omega),\circ))$ such that for any objective function f, for any constraint g_i, for any $\alpha > 0$, for any state $\mathbf{s} \in \Omega$, and for any $\boldsymbol{u} \in \mathbb{U}^\lambda$,

$$\mathscr{F}^{\left(f(\mathbf{x}),\{g_i(\mathbf{x})\}_{i=1,\cdots,m}\right)}(\mathbf{s},\boldsymbol{u}) = \Phi(1/\alpha)\left(\mathscr{F}^{\left(f(\alpha\mathbf{x}),\{g_i(\alpha\mathbf{x})\}_{i=1,\cdots,m}\right)}(\Phi(\alpha)(\mathbf{s}),\boldsymbol{u})\right) \quad .$$

The next proposition states translation-invariance of Algorithm 1.

Proposition 1. *Algorithm 1 is translation-invariant and the associated group homomorphism Φ is given by*

$$\Phi(\boldsymbol{x}_0)(\boldsymbol{x},\sigma,\gamma,\omega) = (\boldsymbol{x}+\boldsymbol{x}_0,\sigma,\gamma,\omega) \quad , \tag{16}$$

for all $\boldsymbol{x}_0,\boldsymbol{x} \in \mathbb{R}^n$, for all $\sigma \in \mathbb{R}$, and for all $\gamma,\omega \in \mathbb{R}^m$.

The proof of this proposition is given in Appendix A.1. In the next proposition we state the scale-invariance of Algorithm 1 under scale-invariance of the transition function $\mathscr{G}_\sigma$.

Proposition 2. *If the update function $\mathscr{G}_\sigma$ of the step-size satisfies the following condition*

$$\mathscr{G}_\sigma(\sigma_t,\varsigma * \boldsymbol{U}_{t+1}) = \alpha\mathscr{G}_\sigma(\sigma_t/\alpha,\varsigma * \boldsymbol{U}_{t+1}) \quad , \tag{17}$$

for all $\alpha > 0$, then Algorithm 1 is scale-invariant and the associated group homomorphism Φ is defined as

$$\Phi(\alpha)(\boldsymbol{x},\sigma,\gamma,\omega) = (\boldsymbol{x}/\alpha,\sigma/\alpha,\gamma,\omega) \quad , \tag{18}$$

for all $\alpha > 0$, for all $\boldsymbol{x} \in \mathbb{R}^n$, for all $\sigma \in \mathbb{R}$, and for all $\gamma,\omega \in \mathbb{R}^m$.

The proof of the proposition is given in Appendix A.2.

In the next section, we illustrate how translation and scale-invariance induce the existence of a homogeneous Markov chain whose stability implies linear convergence.

7 Analysis

In this section, we demonstrate the existence of an underlying homogeneous Markov chain to Algorithm 1, given the augmented Lagrangian in (10) satisfies a particular condition. To construct the Markov chain, we exploit invariance properties of Algorithm 1, as well as the updates of the Lagrange factors and the penalty factors.

As stated in Section 4, we assume that the optimization problem admits a unique global optimum $\mathbf{x}_{\mathrm{opt}}$ and that the constraints g_i, $i = 1,\cdots,m$, satisfy the LICQ at $\mathbf{x}_{\mathrm{opt}}$, hence that the vector γ_{opt} of Lagrange multipliers is unique. Once we have the Markov chain, we show how its stability leads to linear convergence of (i) the current solution $\mathbf{X}_t$ towards the optimum $\mathbf{x}_{\mathrm{opt}}$, (ii) the vector of Lagrange factors γ_t towards the vector of Lagrange multipliers γ_{opt}, and (iii) the step-size σ_t towards 0.

7.1 Homogeneous Markov Chain

We start by recalling the definition of positive homogeneity.

Definition 5. [Definition 4 from [3]] A function $p : X \to Y$ is positive homogeneous of degree $k > 0$ with respect to $\mathbf{x}^* \in X$ if for all $\alpha > 0$ and for all $\mathbf{x} \in X$,

$$p(\mathbf{x}^* + \alpha\mathbf{x}) = \alpha^k p(\mathbf{x}^* + \mathbf{x}) \quad . \tag{19}$$

Example 1. Our linear constraint functions, $g_i(\mathbf{x}) = \mathrm{b}_i^\intercal \mathbf{x} + c_i$, are positive homogeneous of degree 1 with respect to any $\mathbf{x}^* \in \mathbb{R}^n$ that satisfies $g_i(\mathbf{x}^*) = 0$. Indeed,

$$\begin{aligned} g_i(\mathbf{x}^* + \alpha\mathbf{x}) &= \mathrm{b}_i^\intercal(\mathbf{x}^* + \alpha\mathbf{x}) + c_i = \alpha(\mathrm{b}_i^\intercal \mathbf{x}^* + c_i) + \alpha \mathrm{b}_i^\intercal \mathbf{x} \\ &= \alpha g_i(\mathbf{x}^* + \mathbf{x}) \quad , \text{ for all } \alpha > 0. \end{aligned} \tag{20}$$

The following theorem gives sufficient conditions under which the sequence $(\Phi_t)_{t\in\mathbb{N}}$, with $\Phi_t = (\mathbf{Y}_t, \Gamma_t, \omega_t)$, is a homogeneous Markov chain, where the random variables $\mathbf{Y}_t$ and Γ_t are defined in (21) below.

Theorem 1. *Consider the $(\mu/\mu_W, \lambda)$-ES with augmented Lagrangian constraint handling minimizing the augmented Lagrangian h in (10), such that the step-size update function $\mathscr{G}_\sigma$ satisfies the condition in (17). Let $(\boldsymbol{X}_t, \sigma_t, \gamma_t, \omega_t)_{t\in\mathbb{N}}$ be the Markov chain associated to this ES and let $(\boldsymbol{U}_t)_{t\in\mathbb{N}}$ be a sequence of i.i.d. normal vectors. Let $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ such that $g_i(\bar{\boldsymbol{x}}) = 0$ for all $i = 1, \ldots, m$, and let $\bar{\gamma} \in \mathbb{R}^m$. Let*

$$\boldsymbol{Y}_t = \frac{\boldsymbol{X}_t - \bar{\boldsymbol{x}}}{\sigma_t} \quad \text{and} \quad \Gamma_t = \frac{\gamma_t - \bar{\gamma}}{\sigma_t} \quad . \tag{21}$$

Then, if the function $\mathscr{D}h_{\bar{\boldsymbol{x}},\bar{\gamma},\omega} : \mathbb{R}^{n+m} \to \mathbb{R}$ defined as

$$\mathscr{D}h_{\bar{\boldsymbol{x}},\bar{\gamma},\omega}(\boldsymbol{x}, \gamma) = h(\boldsymbol{x}, \gamma, \omega) - h(\bar{\boldsymbol{x}}, \bar{\gamma}, \omega) \tag{22}$$

is positive homogeneous of degree 2 with respect to $[\bar{\boldsymbol{x}}, \bar{\gamma}]$, then the sequence $(\Phi_t)_{t\in\mathbb{N}}$, where $\Phi_t = (\boldsymbol{Y}_t, \Gamma_t, \omega_t)$, is a homogeneous Markov chain that can be defined independently of $(\boldsymbol{X}_t, \sigma_t, \gamma_t, \omega_t)$ as $\boldsymbol{Y}_0 = (\boldsymbol{X}_0 - \bar{\boldsymbol{x}})/\sigma_0$, $\Gamma_0 = (\gamma_0 - \bar{\gamma})/\sigma_0$ and for all t

$$\boldsymbol{Y}_{t+1} = \mathscr{G}_{\boldsymbol{x}}((\boldsymbol{Y}_t, 1), \varsigma * \boldsymbol{U}_{t+1}) / \mathscr{G}_\sigma(1, \varsigma * \boldsymbol{U}_{t+1}) \quad , \tag{23}$$

$$\Gamma_{t+1}^i = \mathscr{H}_\gamma^{g_i(\cdot+\bar{\boldsymbol{x}})}(\Gamma_t^i, \omega_t^i, \tilde{\boldsymbol{Y}}_{t+1}) / \mathscr{G}_\sigma(1, \varsigma * \boldsymbol{U}_{t+1}) \quad , \tag{24}$$

$$\omega_{t+1}^i = \mathscr{H}_\omega^{(f(\cdot+\bar{\boldsymbol{x}}), g_i(\cdot+\bar{\boldsymbol{x}}))}(\omega_t^i, \Gamma_t^i + \bar{\gamma}^i, \tilde{\boldsymbol{Y}}_{t+1}) \quad , \tag{25}$$

with

$$\varsigma = Ord(h(\boldsymbol{Y}_t + \boldsymbol{U}_{t+1}^i + \bar{\boldsymbol{x}}, \Gamma_t + \bar{\gamma}, \omega_t)_{i=1,\cdots,\lambda}) \quad , \tag{26}$$

$$\tilde{\boldsymbol{Y}}_{t+1} = \mathscr{G}_{\boldsymbol{x}}((\boldsymbol{Y}_t, 1), \varsigma * \boldsymbol{U}_{t+1}) \quad , \tag{27}$$

where the Ord operator extracts the permutation of ordered candidate solutions (see (5)).

The proof of Theorem 1 is given in Appendix A.3. The key idea in the proof is that when $\mathscr{D}h_{\bar{\mathbf{x}},\bar{\gamma},\omega_t}$ is positive homogeneous of degree 2 with respect to $[\bar{\mathbf{x}},\bar{\gamma}]$, the same permutation ς is obtained when ranking candidate solutions $\mathbf{X}_t+\sigma_t\mathbf{U}_{t+1}^i$ on $h(\cdot,\gamma_t,\omega_t)$ than when ranking candidate solutions $\mathbf{Y}_t+\mathbf{U}_{t+1}^i$ on $h(\cdot+\bar{\mathbf{x}},\Gamma_t+\bar{\gamma},\omega_t)$, i.e.,

$$\varsigma_{(\mathbf{X}_t,\sigma_t)}^{h(\cdot,\gamma_t,\omega_t)}=\varsigma_{(\mathbf{Y}_t,1)}^{h(\cdot+\bar{\mathbf{x}},\Gamma_t+\bar{\gamma},\omega_t)}=\varsigma \quad .$$

Scale-invariance of Algorithm 1, induced by the property of $\mathscr{G}_\sigma$ in (17), is also used explicitly in the proof while translation-invariance is used implicitly.

Theorem 1 holds for any $\bar{\mathbf{x}}\in\mathbb{R}^n$ such that $g_i(\bar{\mathbf{x}})=0$, for all $i\in\{1,\cdots,m\}$, and for any $\bar{\gamma}\in\mathbb{R}^m$. In particular, it holds for the optimum $\mathbf{x}_{\text{opt}}$ of our constrained problem and the associated vector γ_{opt} of Lagrange multipliers.

The following corollary states that on convex quadratic functions, $(\Phi_t)_{t\in\mathbb{N}}$ (defined in Theorem 1) is a homogeneous Markov chain for $\bar{\mathbf{x}}=\mathbf{x}_{\text{opt}}$ and $\bar{\gamma}=\gamma_{\text{opt}}$.

Corollary 1. *Let $(\boldsymbol{X}_t,\sigma_t,\gamma_t,\omega_t)_{t\in\mathbb{N}}$ be the Markov chain associated to the $(\mu/\mu_W,\lambda)$-ES in 1 optimizing the augmented Lagrangian h in (10), with f convex quadratic defined as*

$$f(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} \quad , \tag{28}$$

where $\boldsymbol{H}\in\mathbb{R}^{n\times n}$ is a symmetric positive-definite matrix. Let $\boldsymbol{Y}_t=\frac{\boldsymbol{X}_t-\boldsymbol{x}_{opt}}{\sigma_t}$ and $\Gamma_t=\frac{\gamma_t-\gamma_{\text{opt}}}{\sigma_t}$, where x_{opt} is the optimum of the constrained problem and γ_{opt} is the vector of the associated Lagrange multipliers. Then $(\Phi_t)_{t\in\mathbb{N}}$, with $\Phi_t=(\boldsymbol{Y}_t,\Gamma_t,\omega_t)$, is a homogeneous Markov chain defined independently of $(\boldsymbol{X}_t,\sigma_t,\gamma_t,\omega_t)$ as in (23), (24), (25), (26), and (27) by taking $\bar{\boldsymbol{x}}=\boldsymbol{x}_{opt}$ and $\bar{\gamma}=\gamma_{\text{opt}}$.

We prove the corollary by showing that the function $\mathscr{D}h_{\mathbf{x}_{\text{opt}},\gamma_{\text{opt}},\omega}$ defined in (22) is positive homogeneous of degree 2 with respect to $[\mathbf{x}_{\text{opt}},\gamma_{\text{opt}}]$ for $f(\mathbf{x})=\frac{1}{2}\mathbf{x}^\intercal\mathbf{H}\mathbf{x}$. For the proof (see Appendix A.4), we use the following elements:

- The definitions of the gradients of f and g_i, $\nabla_{\mathbf{x}}f(\mathbf{y})=\mathbf{y}^\intercal\mathbf{H}$ and $\nabla_{\mathbf{x}}g_i(\mathbf{y})=\mathrm{b}_i^\intercal$, respectively.
- The KKT stationarity condition at the optimum $\mathbf{x}_{\text{opt}}$

$$\nabla_{\mathbf{x}}f(\mathbf{x}_{\text{opt}})+\sum_{i=1}^m\gamma^i\nabla_{\mathbf{x}}g_i(\mathbf{x}_{\text{opt}})=\mathbf{0} \quad . \tag{29}$$

Remark 2. For a convex quadratic objective function f and linear constraints g_i, $i=1,\cdots,m$, KKT conditions are sufficient conditions for optimality. That is, a point that satisfies KKT conditions is also an optimum of the constrained problem (see [15, Theorem 16.4]). The optimization problem we consider is unimodal, therefore $\mathbf{x}_{\text{opt}}$ is the only point satisfying the KKT conditions.

7.2 Sufficient Conditions for Linear Convergence

In the sequel, we investigate linear convergence of Algorithm 1. There exist many definitions—not always equivalent—of linear convergence. We consider here the almost sure linear convergence whose definition is given in Definition 1. We will also briefly discuss another definition of linear convergence that considers the expected log-progress $\ln \frac{\|\mathbf{X}_{t+1}-\mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_t-\mathbf{x}_{\text{opt}}\|}$.

We start by giving the definitions of an invariant probability measure and positivity [14]. We consider a Markov chain $(\mathbf{X}_t)_{t\in\mathbb{N}}$ that takes its values in a set $\mathscr{X} \subset \mathbb{R}^n$ equipped with its Borel σ-algebra $\mathscr{B}(\mathscr{X})$. The transition probabilities are given by the transition probability kernel P such that for $\mathbf{x} \in \mathscr{X}$ and $B \in \mathscr{B}(\mathscr{X})$

$$P(\mathbf{x}, B) = \Pr(\mathbf{X}_{t+1} \in B \mid \mathbf{X}_t = \mathbf{x}) \ .$$

Definition 6. Let π be a probability measure on $\mathscr{X}$ and let $\mathbf{X}_t \sim \pi$. We say that π is invariant if

$$\pi(B) = \int_{\mathscr{X}} \pi(d\mathbf{x}) P(\mathbf{x}, B) \ .$$

We say that a Markov chain is positive if there exists an invariant probability measure for this Markov chain.

Harris-recurrence [14] is related to the notion of irreducibility. Informally, a Markov chain is φ-irreducible if there exists a nonzero measure φ on $\mathscr{X}$ such that all φ-positive sets (that is, sets $B \in \mathscr{B}(\mathscr{X})$ such that $\varphi(B) > 0$) are reachable from anywhere in $\mathscr{X}$. In such a case, there exists a maximal irreducibility measure ψ that dominates other irreducibility measures [14].

Definition 7. Let $(\mathbf{X}_t)_{t\in\mathbb{N}}$ be a ψ-irreducible Markov chain. A measurable set $B \in \mathscr{B}(\mathscr{X})$ is Harris-recurrent if

$$\Pr\Big(\sum_{t\in\mathbb{N}_>} \mathbf{1}_{\{\mathbf{X}_t\in B\}} = \infty \mid \mathbf{X}_0 = \mathbf{x}\Big) = 1 \ ,$$

for all $\mathbf{x} \in B$. By extension, we say that $(\mathbf{X}_t)_{t\in\mathbb{N}}$ is Harris-recurrent if all ψ-positive sets are Harris-recurrent.

We can now recall Theorem 17.0.1 from [14] that gives sufficient conditions for the application of a LLN for Markov chains.

Theorem 2 (Theorem 17.0.1 from [14])**.** *Let $\mathbf{Z}$ be a positive Harris-recurrent chain with invariant probability π. Then the LLN holds for any function q such that $\pi(|q|) = \int |q(z)|\pi(dz) < \infty$, that is, for any initial state $\mathbf{Z}_0$, $\lim_{t\to\infty} \frac{1}{t}\sum_{k=0}^{t-1} q(\mathbf{Z}_k) = \pi(q)$ almost surely.*

Consider now Algorithm 1 minimizing the augmented Lagrangian h in (10) corresponding to the optimization problem in (8), such that the function $\mathscr{D}h_{\mathbf{x}_{\text{opt}},\gamma_{\text{opt}},\omega_t}$ defined in (22) is positive homogeneous of degree 2 with respect to $[\mathbf{x}_{\text{opt}}, \gamma_{\text{opt}}]$. By virtue of Theorem 1, $(\Phi_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain. The following theorem gives sufficient conditions under which Algorithm 1 converges to the optimum $\mathbf{x}_{\text{opt}}$ of the constrained problem, as well as to the corresponding Lagrange multiplier γ_{opt}.

Theorem 3. *Let $(X_t, \sigma_t, \gamma_t, \omega_t)_{t\in\mathbb{N}}$ be the Markov chain associated to Algorithm 1 optimizing the augmented Lagrangian h such that the function $\mathscr{D}h_{\boldsymbol{x}_{opt},\gamma_{\text{opt}},\omega_t}$ defined in (22) is positive homogeneous of degree 2 with respect to $[\boldsymbol{x}_{opt}, \gamma_{\text{opt}}]$, where $\boldsymbol{x}_{opt}$ is the optimum of the constrained problem (8) and γ_{opt} is the corresponding Lagrange multiplier. Let $(\Phi_t)_{t\in\mathbb{N}}$ be the Markov chain defined in Theorem 1 and assume that it is positive Harris-recurrent with invariant probability measure π, that $E_\pi(|\ln\|[\phi]_1\|\|) < \infty$, $E_\pi(|\ln\|[\phi]_2\|\|) < \infty$, and $E_\pi(\mathscr{R}(\phi)) < \infty$, where*

$$\mathscr{R}(\phi) = E\left(\ln(\mathscr{G}_\sigma(1, \varsigma * \boldsymbol{U}_{t+1}))|\Phi_t = \phi\right) \ . \tag{30}$$

Then for all $\boldsymbol{X}_0$, for all σ_0, for all γ_0, and for all ω_0,

$$\lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\boldsymbol{X}_t - \boldsymbol{x}_{opt}\|}{\|\boldsymbol{X}_0 - \boldsymbol{x}_{opt}\|} = \lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\gamma_t - \gamma_{\text{opt}}\|}{\|\gamma_0 - \gamma_{\text{opt}}\|} = \lim_{t\to\infty} \frac{1}{t} \ln \frac{\sigma_t}{\sigma_0} = -CR \ a.s. \ ,$$

where

$$-CR = \int \mathscr{R}(\phi)\pi(d\phi) \ .$$

The proof idea is similar to the one discussed in Section 3 for the unconstrained case, where the quantities $\frac{1}{t}\ln\frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|}$, $\frac{1}{t}\ln\frac{\|\gamma_t - \gamma_{\text{opt}}\|}{\|\gamma_0 - \gamma_{\text{opt}}\|}$, and $\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0}$ are expressed as a function of the Markov chain Φ_t. The detailed proof of Theorem 1 is given in Appendix A.5.

While in the previous theorem we have presented sufficient conditions on the Markov chain Φ_t for the almost sure linear convergence of the algorithm, other sufficient conditions can allow to derive the geometric convergence of the expected log-progress. Typically, assuming we have proven a so-called geometric drift for the chain Φ_t, plus some assumptions ensuring that the conditional log-progress is dominated by the drift function (see for instance [7, Theorem 5.4]), then

$$\sum_t r^t \left| E_{\Phi_0} \ln \frac{\|\mathbf{X}_{t+1} - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|} - (-\text{CR})\right| \leq RV(\Phi_0) \ , \tag{31}$$

where $r > 1$, R is a positive constant and $V \geq 1$ is the drift function. Equation (31) also holds when replacing $\ln\frac{\|\mathbf{X}_{t+1} - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}$ by $\ln\frac{\|\gamma_{t+1} - \gamma_{\text{opt}}\|}{\|\gamma_t - \gamma_{\text{opt}}\|}$ and $\ln\frac{\sigma_{t+1}}{\sigma_t}$.

8 Empirical Results

We describe here our experimental setting and discuss the obtained results.

8.1 Step-Size Adaptation Mechanism

We test Algorithm 1 with cumulative step-size adaptation (CSA) [12]. The idea of CSA consists in keeping track of the successive steps taken by the algorithm in the search space.

This is done by computing an evolution path, $\boldsymbol{p}_t^\sigma$, according to

$$\boldsymbol{p}_{t+1}^\sigma = (1-c_\sigma)\boldsymbol{p}_t^\sigma + \sqrt{\frac{c_\sigma(2-c_\sigma)}{\sum_{k=1}^\mu w_k^2}} \sum_{k=1}^\mu w_k \mathbf{U}_{t+1}^{\varsigma(k)} \quad , \tag{32}$$

where $0 < c_\sigma \leq 1$ and $\boldsymbol{p}_0^\sigma = \mathbf{0}$. The constant $\sqrt{\frac{c_\sigma(2-c_\sigma)}{\sum_{k=1}^\mu w_k^2}}$ is a normalization factor that is chosen such that under random selection, if $\boldsymbol{p}_t^\sigma$ is normally distributed ($\boldsymbol{p}_t^\sigma \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$), then $\boldsymbol{p}_{t+1}^\sigma$ is identically distributed [10, 11]. The evolution path is used to adapt the step-size σ_t according to the following rule.

$$\sigma_{t+1} = \sigma_t \exp^{\frac{c_\sigma}{d_\sigma}} \left(\frac{\|\boldsymbol{p}_{t+1}^\sigma\|}{E\|\mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})\|} - 1 \right) \quad . \tag{33}$$

The norm of the evolution path is compared to the expected norm of a standard normal vector by computing the ratio $\frac{\|\boldsymbol{p}_{t+1}^\sigma\|}{E\|\mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})\|}$ and the step-size is updated depending on this ratio: if $\frac{\|\boldsymbol{p}_{t+1}^\sigma\|}{E\|\mathcal{N}(\mathbf{0},\mathbf{I}_{n\times n})\|} \geq 1$, σ_t is increased as this suggests that the progress is too slow. Otherwise, σ_t is decreased. d_σ is a damping factor whose role is to moderate the changes in σ_t values.

In order for this adaptation mechanism to be compliant with our general adaptation rule $\mathscr{G}_\sigma(\sigma_t, \varsigma * \mathbf{U}_{t+1})$ (see (13)), we take $c_\sigma = 1$, that is, we consider CSA without cumulation. In this case, (32) becomes

$$\boldsymbol{p}_{t+1}^\sigma = \sqrt{\frac{1}{\sum_{k=1}^\mu w_k^2}} \sum_{k=1}^\mu w_k \mathbf{U}_{t+1}^{\varsigma(k)} \quad .$$

For the damping factor, we use

$$d_\sigma = 2 + 2\max\left(0, \sqrt{\frac{1/\sum_{k=1}^\mu w_k^2 - 1}{n+1}} - 1\right) \quad ,$$

which is the default value recommended in [11] with $c_\sigma = 1$.

8.2 Simulations of the Markov Chain and Single Runs

We test Algorithm 1 on two convex quadratic functions, as a particular case of Corollary 1: the sphere function, f_{sphere}, and the ellipsoid function, $f_{\text{ellipsoid}}$, with a moderate condition number. They are defined according to (28) by taking (i) $\mathbf{H} = \mathbf{I}_{n\times n}$ for f_{sphere} and (ii) $\mathbf{H}$ diagonal with diagonal elements $[\mathbf{H}]_i = \alpha^{\frac{i-1}{n-1}}$, $i = 1, \cdots, n$, for $f_{\text{ellipsoid}}$ and with a condition number $\alpha = 10$.

We choose $\mathbf{x}_{\text{opt}}$ to be at $(10, \cdots, 10)^\intercal$ and construct the (active) constraints following the steps below:

- For the first constraint, $\mathrm{b}_1 = -\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}})^\intercal$ and $c_1 = -\mathrm{b}_1^\intercal \mathbf{x}_{\text{opt}}$,
- For the $m-1$ remaining constraints, we choose the constraint normal b_i as a standard normal vector ($\mathrm{b}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$) and $c_i = -\mathrm{b}_i^\intercal \mathbf{x}_{\text{opt}}$. We choose the point $\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}})^\intercal = -\mathrm{b}_1$ to be feasible, along with $\mathbf{x}_{\text{opt}}$. Therefore, if $g_i(\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}})^\intercal) > 0$, we modify b_i and c_i according to: $\mathrm{b}_i = -\mathrm{b}_i$ and $c_i = -c_i$.

With the construction above, the constraints satisfy the LICQ (see Remark 1) with probability one. In such a case, the unique vector of Lagrange multipliers associated to $\mathbf{x}_{\text{opt}}$ is $\gamma_{\text{opt}} = (1,0,\cdots,0)^\intercal$.

As for the parameters of Algorithm 1, we choose the default values in [11] for both λ and μ. We set the weights w_i, $i = 1,\cdots,\mu$, according to [1], where they are chosen to be optimal on the sphere function in infinite dimension. We take $d_\gamma = d_\omega = 5$, $\chi = 2^{1/n}$, $k_1 = 3$, and $k_2 = 5$.

We run Algorithm 1 and simulate the Markov chain $(\Phi_t)_{t\in\mathbb{N}}$ (defined in Theorem 1) in $n = 10$ on f_{sphere} and $f_{\text{ellipsoid}}$ with $m = 1,2,5,9$ constraints. For each problem, we test three different initial values of the penalty vector $\omega_0 = (1,\cdots,1)^\intercal$, $(10^3,\cdots,10^3)^\intercal$, $(10^{-3},\cdots,10^{-3})^\intercal$. In all the tests, $\mathbf{X}_0$ and $\mathbf{Y}_0$ are sampled uniformly in $[-5,5]^n$, $\sigma_0 = 1$, and $\gamma_0 = \Gamma_0 = (5,\cdots,5)^\intercal$.

Figures 2-5 show simulations of the Markov chain on f_{sphere} (left column) and $f_{\text{ellipsoid}}$ (right column) subject to 1, 2, 5, and 9 constraints respectively. Displayed are the normalized distance to $\mathbf{x}_{\text{opt}}$, $\|\mathbf{Y}_t\|$ (red), the normalized distance to γ_{opt}, $\|\Gamma_t\|$ (green), and the norm of the vector of penalty factors, $\|\omega_t\|$ (blue) in log-scale, for $\omega_0 = (1,\cdots,1)^\intercal$ (first row), $\omega_0 = (10^3,\cdots,10^3)^\intercal$ (second row), and $\omega_0 = (10^{-3},\cdots,10^{-3})^\intercal$ (third row). We observe an overall convergence to a stationary distribution, independently of ω_0, after a certain number of iterations. For $\omega_0 = (10^3,\cdots,10^3)^\intercal$, the adaptation phase before reaching the stationary state is longer than with $\omega_0 = (1,\cdots,1)^\intercal$ or $\omega_0 = (10^{-3},\cdots,10^{-3})^\intercal$ on both f_{sphere} and $f_{\text{ellipsoid}}$. It also increases with increasing m: it takes approximately 4×10^3 iterations on f_{sphere} and $f_{\text{ellipsoid}}$ with $m = 1$ (Figure 2) and approximately 6×10^3 iterations with $m = 9$ (Figure 5). Indeed, the problem becomes more difficult for large m (as shown below with single runs). We also observe from Figures 2-5 that $\|\omega_t\|$ stabilizes around a larger value as m increases (approximately 4×10^{-4} and 6×10^{-5} on f_{sphere} and $f_{\text{ellipsoid}}$ respectively with $m = 1$ versus approximately 1 and 4 with $m = 9$).

Figures 6-9 show single runs of Algorithm 1 on the same constrained problems described previously. Results on constrained f_{sphere} and constrained $f_{\text{ellipsoid}}$ are displayed in left and right columns respectively. The displayed quantities are (i) the distance to the optimum, $\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|$ (red), (ii) the distance to the Lagrange multipliers, $\|\gamma_t - \gamma_{\text{opt}}\|$ (green), (iii) the norm of the penalty vector, $\|\omega_t\|$ (blue), and (iv) the step-size, σ_t (purple), in log-scale. Linear convergence occurs after an adaptation phase whose length depends on the accuracy of the choice of the initial parameters: for $m = 1$ and $\omega_0 = (10^{-3},\cdots,10^{-3})^\intercal$ (Figure 6, third row), linear convergence occurs after only around 30 iterations because ω_0 is already close to a stationary value (see Figure 2). On f_{sphere} with $m = 2$ (Figure 7, left column), the algorithm reaches a distance to $\mathbf{x}_{\text{opt}}$ of 10^{-4} after around 750 iterations with $\omega_0 = (1,\cdots,1)^\intercal$, compared to around 2500 iterations with $\omega_0 = (10^3,\cdots,10^3)^\intercal$ and around 1300 iterations

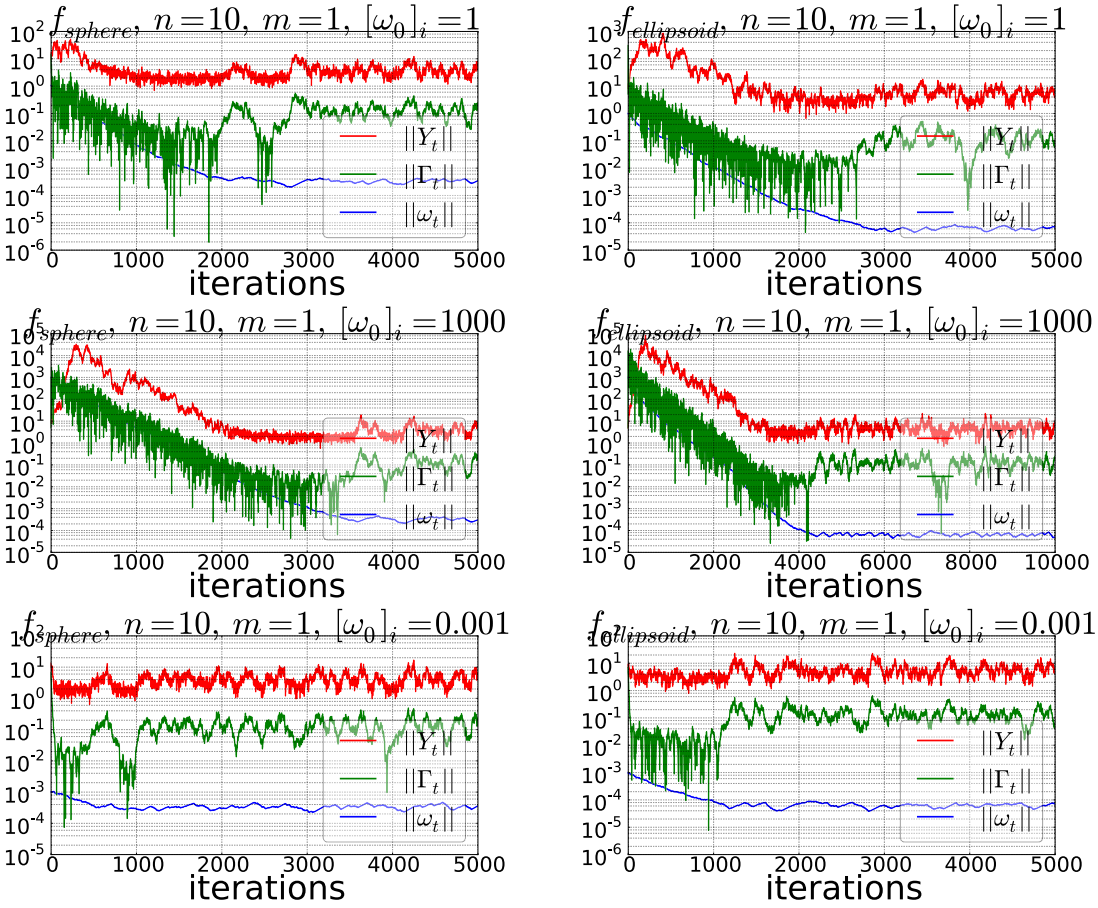

Figure 2: Simulations of the Markov chain on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 1$ in $n = 10$.

with $\omega_0 = (10^{-3}, \cdots, 10^{-3})^\intercal$. The reason is that $\omega_0 = (1, \cdots, 1)^\intercal$ is closer to the stationary value in this case (Figure 3, left column). As the number of constraints increases (Figures 8 and 9), the number of iterations needed to reach a given precision increases: it takes more than 2 times longer to reach a distance from the optimum of 10^{-4} on both f_{sphere} and $f_{\text{ellipsoid}}$ with $m = 9$ and $\omega_0 = (1, \cdots, 1)^\intercal$ (Figure 9, first row) than with $m = 1$ (Figure 6, first row). These results are consistent with the simulations of the Markov chain in that the observed stability of the Markov chain leads to linear convergence of the algorithm—as stated in Theorem 3.

9 Discussion

In this work, we investigated linear convergence of a $(\mu/\mu_W, \lambda)$-ES with an augmented Lagrangian constraint handling on the linearly constrained problem where all the constraints are active. We adopted a Markov chain approach and exhibited a homogeneous Markov chain on problems where the associated augmented Lagrangian, centered in the optimum

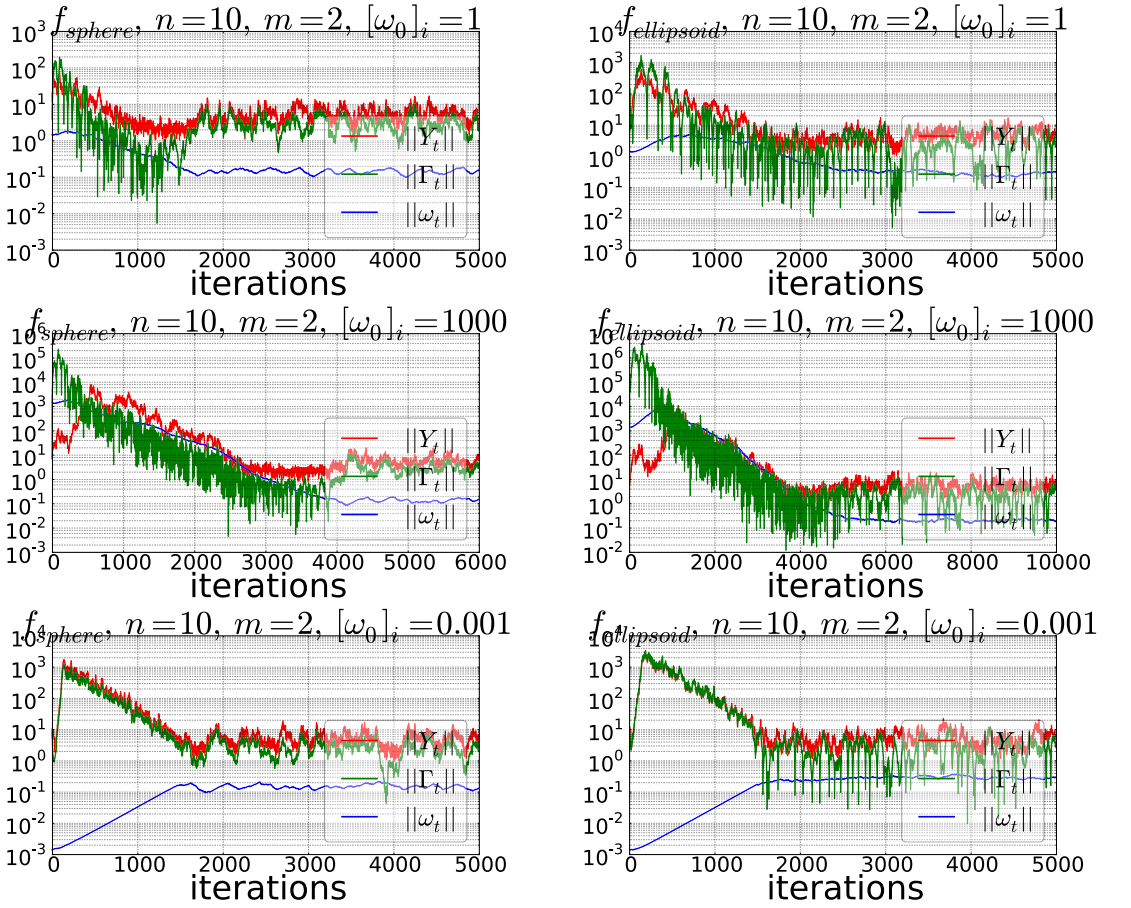

Figure 3: Simulations of the Markov chain on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 2$ in $n = 10$.

and the corresponding Lagrange multipliers, is positive homogeneous of degree 2. We gave sufficient stability conditions which, when satisfied by the Markov chain, lead to linear convergence to the optimum as well as to the Lagrange multipliers. Simulations of the Markov chain on linearly constrained convex quadratic functions (as a particular case of the exhibited class of functions) show empirical evidence of stability for the tested parameter setting. We draw attention, however, to the fact that the observed stability may depend on the chosen parameter setting—in particular the damping factors for the Lagrange factors and the penalty factors—and proper parameter values are necessary to observe stability, especially in larger dimensions and for large numbers of constraints.

The conducted analysis gives insight into the behavior of the practical $(\mu/\mu_W, \lambda)$-ES obtained when optimizing the augmented Lagrangian presented in (9). Indeed, we focus our study on the most difficult case in practice, where all the constraints are active at the optimum.

Finally, this work illustrates how the Markov chain approach—which is already applied to prove linear convergence of randomized optimization algorithms in the unconstrained case—can be extended to the constrained case.

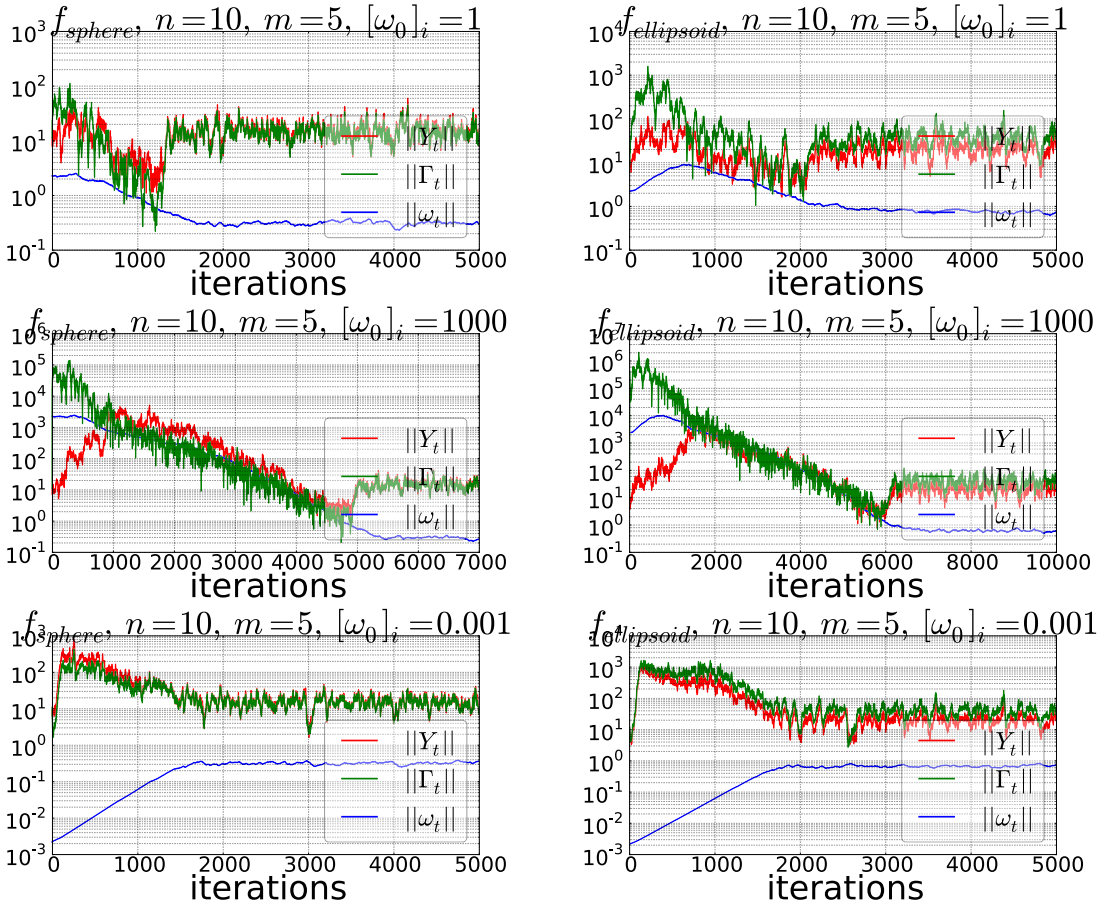

Figure 4: Simulations of the Markov chain on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 5$ in $n = 10$.

Acknowledgments

This work was supported by the grant ANR-2012-MONU-0009 (NumBBO) from the French National Research Agency.

References

[1] D. V. Arnold. Optimal weighted recombination. In *Foundations of Genetic Algorithms*, pages 215–237. Springer, 2005.

[2] D. V. Arnold and J. Porter. Towards an Augmented Lagrangian Constraint Handling Approach for the $(1+1)$-ES. In *Genetic and Evolutionary Computation Conference*, pages 249–256. ACM Press, 2015.

[3] A. Atamna, A. Auger, and N. Hansen. Analysis of Linear Convergence of a $(1+1)$-ES with Augmented Lagrangian Constraint Handling. In *Genetic and Evolutionary Computation Conference*, pages 213–220. ACM Press, 2016.

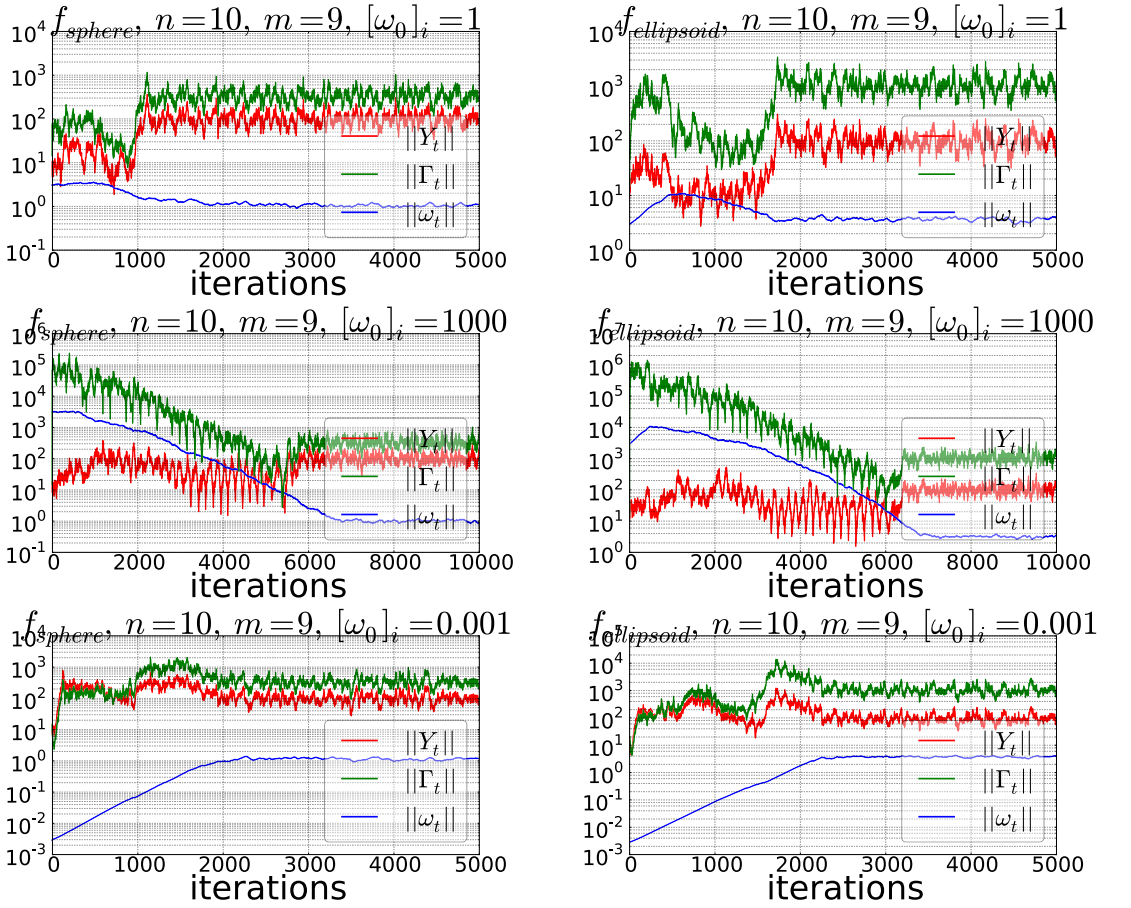

Figure 5: Simulations of the Markov chain on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 9$ in $n = 10$.

[4] A. Atamna, A. Auger, and N. Hansen. Augmented Lagrangian Constraint Handling for CMA-ES—Case of a Single Linear Constraint. In *Parallel Problem Solving from Nature*, pages 181–191. Springer, 2016.

[5] A. Auger. Convergence Results for the $(1,\lambda)$-SA-ES Using the Theory of φ-Irreducible Markov Chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[6] A. Auger and N. Hansen. Linear Convergence on Positively Homogeneous Functions of a Comparison Based Step-Size Adaptive Randomized Search: the $(1+1)$ ES with Generalized One-Fifth Success Rule. Submitted for publication, 2013.

[7] A. Auger and N. Hansen. Linear Convergence of Comparison-Based Step-Size Adaptive Randomized Search via Stability of Markov Chains. *SIAM Journal on Optimization*, 26(3):1589–1624, 2016.

[8] A. Bienvenüe and O. François. Global Convergence of Evolution Strategies in Spherical Problems: Some Simple Proofs and Difficulties. *Theoretical Computer Science*, 306(1–3):269–289, 2003.

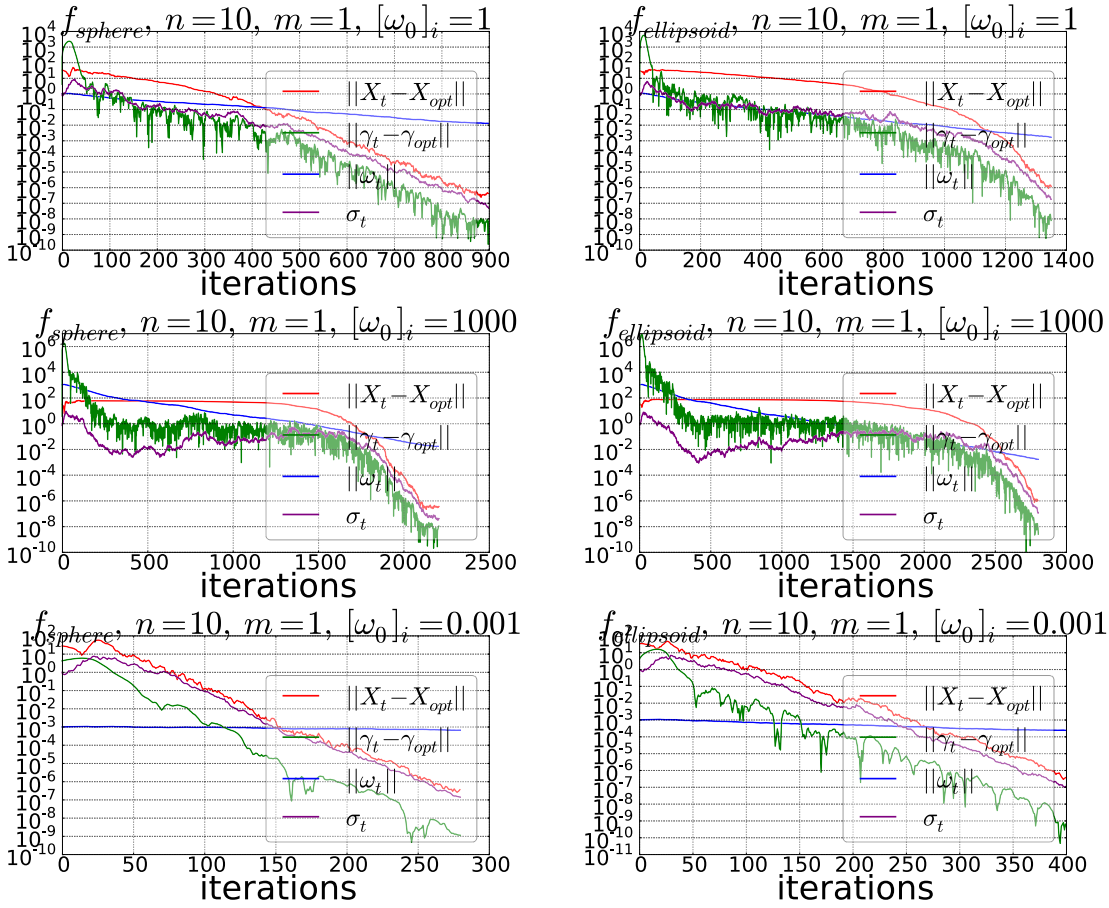

Figure 6: Single runs on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 1$ in $n = 10$, with three different values of ω_0.

[9] K. Deb and S. Srivastava. A Genetic Algorithm Based Augmented Lagrangian Method for Constrained Optimization. *Computational Optimization and Applications*, 53(3):869–902, 2012.

[10] H. Hansen, D. V. Arnold, and A. Auger. Evolution strategies. In J. Kacprzyk and W. Pedrycz, editors, *Handbook of Computational Intelligence*, chapter 44, pages 871–898. Springer, 2015.

[11] N. Hansen. The CMA Evolution Strategy: A Tutorial. http://arxiv.org/pdf/1604.00772v1.pdf, 2016.

[12] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[13] M. R. Hestenes. Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

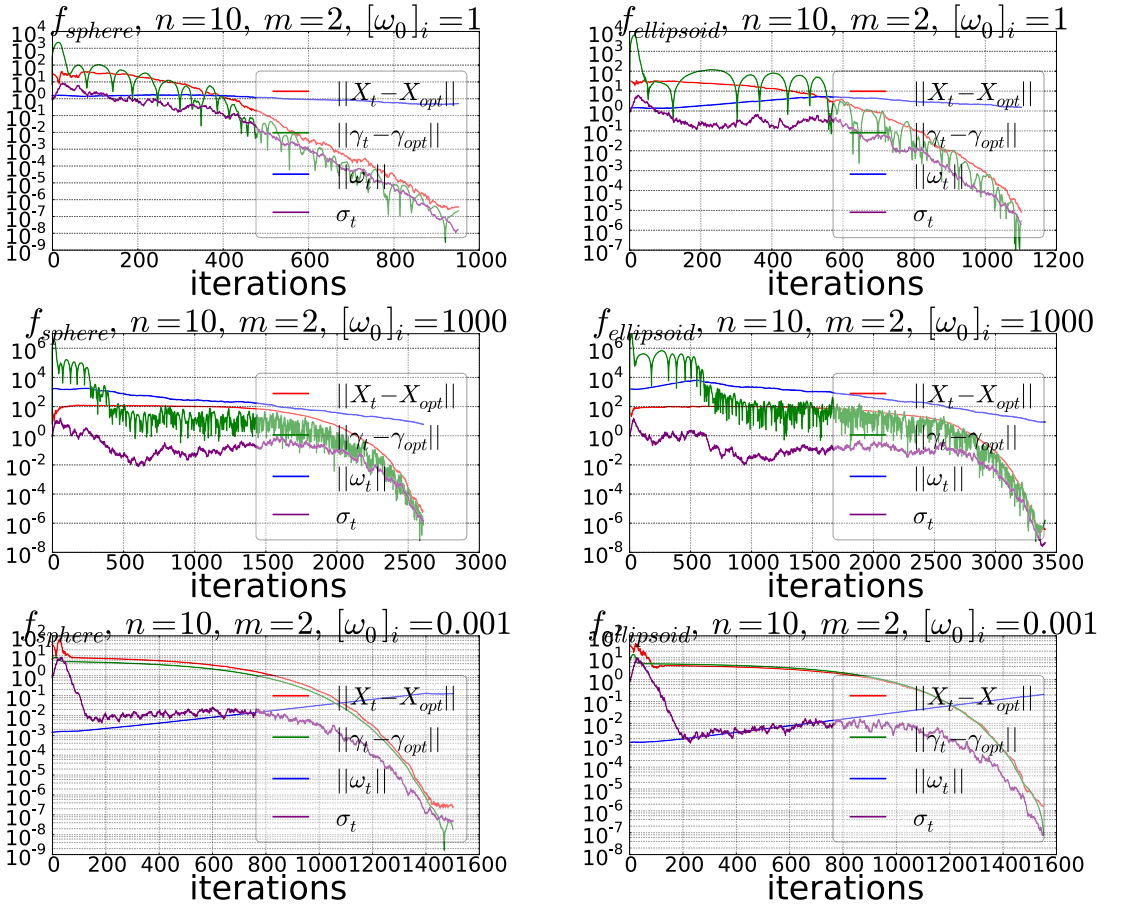

Figure 7: Single runs on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 2$ in $n = 10$, with three different values of ω_0.

[14] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, 1993.

[15] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[16] M. J. D. Powell. A Method for Nonlinear Constraints in Minimization Problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, 1969.

[17] M.-J. Tahk and B.-C. Sun. Coevolutionary Augmented Lagrangian Methods for Constrained Optimization. *IEE Transactions on Evolutionary Computation*, 4(2):114–124, 2000.

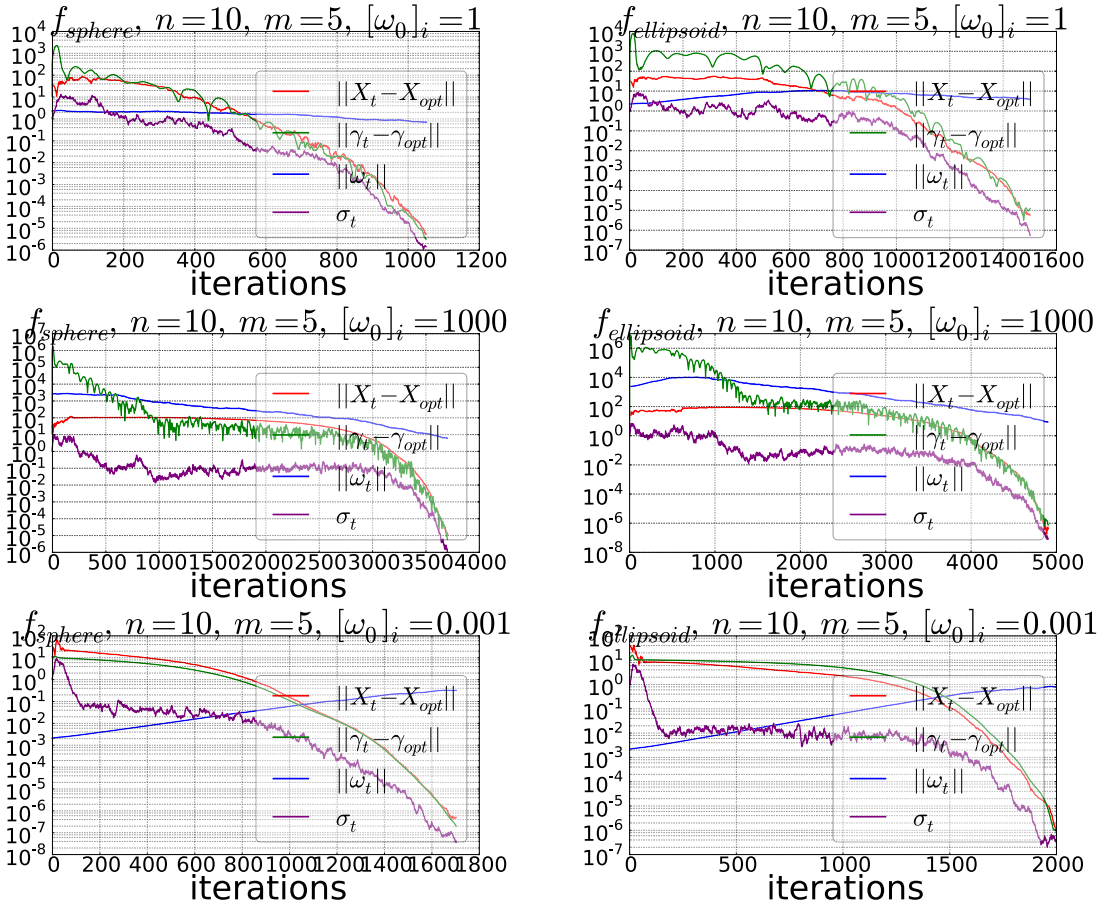

Figure 8: Single runs on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 5$ in $n = 10$, with three different values of ω_0.

A Proofs

A.1 Proof of Proposition 1

For Algorithm 1, the state $\mathbf{s}_t = (\mathbf{X}_t, \sigma_t, \gamma_t, \omega_t)$. Let

$$(\mathbf{X}'_{t+1}, \sigma'_{t+1}, \gamma'_{t+1}, \omega'_{t+1}) = \mathscr{F}^{(f(\cdot-\mathbf{x}_0), \{g_i(\cdot-\mathbf{x}_0)\}_{i=1,\cdots,m})}(\Phi(\mathbf{x}_0)(\mathbf{X}_t, \sigma_t, \gamma_t, \omega_t), \mathbf{U}_{t+1}) \ .$$

Given the definition of $\Phi(\mathbf{x}_0)$ in (16) and the update functions $\mathscr{G}_{\mathbf{x}}$, $\mathscr{G}_\sigma$, $\mathscr{H}_\gamma$, and $\mathscr{H}_\omega$ in (12), (13), (14), and (15) respectively, we have

$$\begin{aligned}
\mathbf{X}'_{t+1} &= \mathscr{G}_{\mathbf{x}}((\mathbf{X}_t + \mathbf{x}_0, \sigma_t), \varsigma^{h(\cdot-\mathbf{x}_0,\gamma_t,\omega_t)}_{(\mathbf{X}_t+\mathbf{x}_0,\sigma_t)} * \mathbf{U}_{t+1}) \\
&= \mathbf{X}_t + \mathbf{x}_0 + \sigma_t \sum_{i=1}^{\mu} w_i [\varsigma^{h(\cdot-\mathbf{x}_0,\gamma_t,\omega_t)}_{(\mathbf{X}_t+\mathbf{x}_0,\sigma_t)} * \mathbf{U}_{t+1}]_i \ , \\
\sigma'_{t+1} &= \mathscr{G}_\sigma(\sigma_t, \varsigma^{h(\cdot-\mathbf{x}_0,\gamma_t,\omega_t)}_{(\mathbf{X}_t+\mathbf{x}_0,\sigma_t)}) \ .
\end{aligned}$$

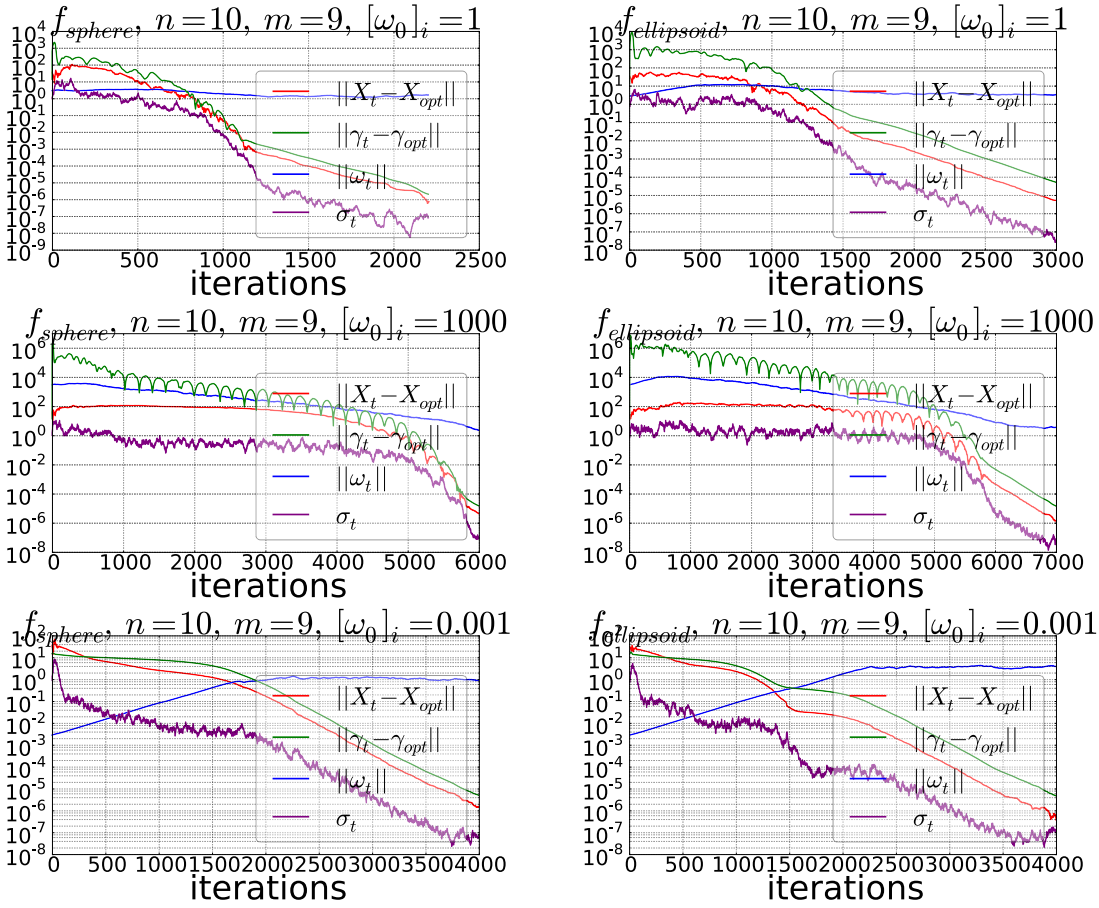

Figure 9: Single runs on f_{sphere} (left) and $f_{\text{ellipsoid}}$ (right) with $m = 9$ in $n = 10$, with three different values of ω_0.

On the other hand, we have

$$\varsigma^{h(\cdot-\mathbf{x}_0,\gamma_t,\omega_t)}_{(\mathbf{X}_t+\mathbf{x}_0,\sigma_t)} = Ord(h(\mathbf{X}_t+\mathbf{x}_0+\sigma_t\mathbf{U}^i_{t+1}-\mathbf{x}_0,\gamma_t,\omega_t)_{i=1,\cdots,\lambda}) = \varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)} \quad .$$

It follows that

$$\begin{aligned}\mathbf{X}'_{t+1} &= \mathscr{G}_{\mathbf{x}}((\mathbf{X}_t,\sigma_t),\varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)} * \mathbf{U}_{t+1}) + \mathbf{x}_0 = \mathbf{X}_{t+1}+\mathbf{x}_0 \quad , \\ \sigma'_{t+1} &= \mathscr{G}_{\sigma}(\sigma_t,\varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)}) = \sigma_{t+1} \quad .\end{aligned} \tag{34}$$

Using (34), we obtain

$$\begin{aligned}\gamma'^i_{t+1} &= \mathscr{H}^{g_i(\cdot-\mathbf{x}_0)}_{\gamma}(\gamma^i_t,\omega^i_t,\mathbf{X}'_{t+1}) = \gamma^i_t + \frac{\omega^i_t}{d_\gamma} g_i(\mathbf{X}'_{t+1}-\mathbf{x}_0) \\ &= \mathscr{H}^{g_i}_{\gamma}(\gamma^i_t,\omega^i_t,\mathbf{X}_{t+1}) = \gamma^i_{t+1},\ i=1,\cdots,m \quad ,\end{aligned}$$

$$\omega'^{i}_{t+1} = \mathscr{H}_{\omega}^{(f(\cdot-\mathbf{x}_0),g_i(\cdot-\mathbf{x}_0))}(\omega^i_t,\gamma^i_t,\mathbf{X}_t+\mathbf{x}_0,\mathbf{X}'_{t+1})$$
$$= \begin{cases} \omega^i_t \chi^{1/(4d_\omega)} & \text{if } \omega^i_t g_i(\mathbf{X}'_{t+1}-\mathbf{x}_0)^2 < k_1 \frac{|h(\mathbf{X}'_{t+1}-\mathbf{x}_0,\gamma_t,\omega_t)-h(\mathbf{X}_t+\mathbf{x}_0-\mathbf{x}_0,\gamma_t,\omega_t)|}{n} \\ & \text{or } k_2|g_i(\mathbf{X}'_{t+1}-\mathbf{x}_0)-g_i(\mathbf{X}_t+\mathbf{x}_0-\mathbf{x}_0)| < |g_i(\mathbf{X}_t+\mathbf{x}_0-\mathbf{x}_0)| \\ \omega^i_t \chi^{-1/d_\omega} & \text{otherwise, } i=1,\cdots,m \end{cases}$$
$$= \mathscr{H}_{\omega}^{(f,g_i)}(\omega^i_t,\gamma^i_t,\mathbf{X}_t,\mathbf{X}_{t+1}) = \omega^i_{t+1},\ i=1,\cdots,m\ .$$

Therefore,

$$(\mathbf{X}_{t+1}+\mathbf{x}_0,\sigma_{t+1},\gamma_{t+1},\omega_{t+1}) = \mathscr{F}^{(f(\cdot-\mathbf{x}_0),\{g_i(\cdot-\mathbf{x}_0)\}_{i=1,\cdots,m})}(\Phi(\mathbf{x}_0)(\mathbf{X}_t,\sigma_t,\gamma_t,\omega_t),\mathbf{U}_{t+1})\ . \tag{35}$$

By applying the inverse transformation $\Phi(-\mathbf{x}_0)$ to (35), we recover $\mathscr{F}^{(f,\{g_i\}_{i=1,\cdots,m})}(\mathbf{X}_t,\sigma_t,\gamma_t,\omega_t)$.

A.2 Proof of Proposition 2

The state at iteration t is $\mathbf{s}_t = (\mathbf{X}_t,\sigma_t,\gamma_t,\omega_t)$. Let

$$(\mathbf{X}'_{t+1},\sigma'_{t+1},\gamma'_{t+1},\omega'_{t+1}) = \mathscr{F}^{(f(\alpha\cdot),\{g_i(\alpha\cdot)\}_{i=1,\cdots,m})}(\Phi(\alpha)(\mathbf{X}_t,\sigma_t,\gamma_t,\omega_t),\mathbf{U}_{t+1})\ .$$

By definition, we have

$$\varsigma^{h(\alpha\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t/\alpha,\sigma_t/\alpha)} = Ord(h(\alpha(\mathbf{X}_t/\alpha+\sigma_t/\alpha\mathbf{U}^i_{t+1}),\gamma_t,\omega_t)_{i=1,\cdots,\lambda}) = \varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)}\ .$$

Using the definition of $\Phi(\alpha)$ in (18), (12), (13), (14), (15), and the equation above, it follows

$$\mathbf{X}'_{t+1} = \mathscr{G}_{\mathbf{x}}((\mathbf{X}_t/\alpha,\sigma_t/\alpha),\varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)}*\mathbf{U}_{t+1}) = \frac{\mathbf{X}_t}{\alpha}+\frac{\sigma_t}{\alpha}\sum_{i=1}^{\mu} w_i[\varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)}*\mathbf{U}_{t+1}]_i$$
$$= \frac{1}{\alpha}\mathscr{G}_{\mathbf{x}}((\mathbf{X}_t,\sigma_t),\varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)}*\mathbf{U}_{t+1}) = \frac{\mathbf{X}_{t+1}}{\alpha}\ , \tag{36}$$

and $\sigma'_{t+1} = \mathscr{G}_{\sigma}(\sigma_t/\alpha,\varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)}*\mathbf{U}_{t+1})$. Using the scale-invariance property of $\mathscr{G}_{\sigma}$ (see (17)), we obtain

$$\sigma'_{t+1} = \frac{1}{\alpha}\mathscr{G}_{\sigma}(\sigma_t,\varsigma^{h(\cdot,\gamma_t,\omega_t)}_{(\mathbf{X}_t,\sigma_t)}*\mathbf{U}_{t+1}) = \frac{\sigma_{t+1}}{\alpha}\ .$$

Finally, using (36) we get

$$\gamma'^{i}_{t+1} = \mathscr{H}_{\gamma}^{g_i(\alpha\cdot)}(\gamma^i_t,\omega^i_t,\mathbf{X}'_{t+1}) = \gamma^i_t+\frac{\omega^i_t}{d_\gamma}g_i(\alpha\mathbf{X}'_{t+1})$$
$$= \mathscr{H}_{\gamma}^{g_i}(\gamma^i_t,\omega^i_t,\mathbf{X}_{t+1}) = \gamma^i_{t+1},\ i=1,\cdots,m\ ,$$

and

$$\omega'^{i}_{t+1} = \mathscr{H}_{\omega}^{(f(\alpha\cdot),g_i(\alpha\cdot))}(\omega^i_t,\gamma^i_t,\mathbf{X}_t/\alpha,\mathbf{X}'_{t+1})$$
$$= \begin{cases} \omega^i_t \chi^{1/(4d_\omega)} & \text{if } \omega^i_t g_i(\alpha\mathbf{X}'_{t+1})^2 < k_1 \frac{|h(\alpha\mathbf{X}'_{t+1},\gamma_t,\omega_t)-h(\alpha\mathbf{X}_t/\alpha,\gamma_t,\omega_t)|}{n} \\ & \text{or } k_2|g_i(\alpha\mathbf{X}'_{t+1})-g_i(\alpha\mathbf{X}_t/\alpha)| < |g_i(\alpha\mathbf{X}_t/\alpha)| \\ \omega^i_t \chi^{-1/d_\omega} & \text{otherwise, } i=1,\cdots,m \end{cases}$$
$$= \mathscr{H}_{\omega}^{(f,g_i)}(\omega^i_t,\gamma^i_t,\mathbf{X}_t,\mathbf{X}_{t+1}) = \omega^i_{t+1},\ i=1,\cdots,m\ .$$

Therefore,

$$\left(\frac{\mathbf{X}_{t+1}}{\alpha}, \frac{\sigma_{t+1}}{\alpha}, \gamma_{t+1}, \omega_{t+1}\right) = \mathscr{F}^{(f(\alpha\cdot),\{g_i(\alpha\cdot)\}_{i=1,\cdots,m})}(\Phi(\alpha)(\mathbf{X}_t, \sigma_t, \gamma_t, \omega_t), \mathbf{U}_{t+1}) \quad . \tag{37}$$

By applying the inverse transformation $\Phi(1/\alpha)$ to (37), we obtain $\mathscr{F}^{(f,\{g_i\}_{i=1,\cdots,m})}(\mathbf{X}_t, \sigma_t, \gamma_t, \omega_t)$.

A.3 Proof of Theorem 1

We have

$$\mathbf{Y}_{t+1} = \frac{\mathbf{X}_{t+1} - \bar{\mathbf{x}}}{\sigma_{t+1}} = \frac{\mathscr{G}_{\mathbf{x}}((\mathbf{X}_t, \sigma_t), \varsigma * \mathbf{U}_{t+1}) - \bar{\mathbf{x}}}{\mathscr{G}_{\sigma}(\sigma_t, \varsigma * \mathbf{U}_{t+1})} \quad .$$

Using translation-invariance and scale-invariance of Algorithm 1, it follows

$$\mathbf{Y}_{t+1} = \frac{\mathscr{G}_{\mathbf{x}}((\mathbf{Y}_t, 1), \varsigma * \mathbf{U}_{t+1})}{\mathscr{G}_{\sigma}(1, \varsigma * \mathbf{U}_{t+1})} \quad ,$$

with

$$\begin{aligned}
\varsigma &= \varsigma_{(\mathbf{X}_t, \sigma_t)}^{h(\cdot, \gamma_t, \omega_t)} = \varsigma_{(\mathbf{Y}_t, 1)}^{h(\sigma_t \cdot + \bar{\mathbf{x}}, \gamma_t, \omega_t)} = \varsigma_{(\mathbf{Y}_t, 1)}^{h(\sigma_t \cdot + \bar{\mathbf{x}}, \sigma_t \Gamma_t + \bar{\gamma}, \omega_t)} \\
&= Ord(h(\sigma_t(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i) + \bar{\mathbf{x}}, \sigma_t \Gamma_t + \bar{\gamma}, \omega_t)_{i=1,\cdots,\lambda}) \\
&= Ord(\mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}(\sigma_t(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i) + \bar{\mathbf{x}}, \sigma_t \Gamma_t + \bar{\gamma})_{i=1,\cdots,\lambda}) \quad ,
\end{aligned}$$

where $\mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}$ is defined in (22) and $\mathbf{Y}_t$ and Γ_t in (21). By positive homogeneity of $\mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}$, it follows

$$\begin{aligned}
\varsigma &= Ord(\sigma_t{}^2 \mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma})_{i=1,\cdots,\lambda}) \\
&= Ord(\mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma})_{i=1,\cdots,\lambda}) \\
&= Ord(h(\mathbf{Y}_t + [\mathbf{U}_{t+1}]_i + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma}, \omega_t)_{i=1,\cdots,\lambda}) \\
&= \varsigma_{(\mathbf{Y}_t, 1)}^{h(\cdot + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma}, \omega_t)} \quad .
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
\Gamma_{t+1}^i = \frac{\gamma_{t+1}^i - \bar{\gamma}^i}{\sigma_{t+1}} = \frac{\mathscr{H}_{\gamma}^{g_i}(\gamma_t^i, \omega_t^i, \mathbf{X}_{t+1}) - \bar{\gamma}^i}{\mathscr{G}_{\sigma}(\sigma_t, \varsigma * \mathbf{U}_{t+1})} &= \frac{\gamma_t^i + \frac{\omega_t^i}{d_\gamma} g_i(\mathbf{X}_{t+1}) - \bar{\gamma}^i}{\sigma_t \mathscr{G}_{\sigma}(1, \varsigma * \mathbf{U}_{t+1})} \\
&= \frac{\Gamma_t^i + \frac{\omega_t^i}{d_\gamma \sigma_t} g_i(\mathbf{X}_{t+1})}{\mathscr{G}_{\sigma}(1, \varsigma * \mathbf{U}_{t+1})} \quad .
\end{aligned}$$

Using positive homogeneity of g_i with respect to $\bar{\mathbf{x}}$ (see (20)) and the definition of $\tilde{\mathbf{Y}}_{t+1}$ in (27), we have

$$g_i(\mathbf{X}_{t+1}) = g_i(\sigma_{t+1}\mathbf{Y}_{t+1} + \bar{\mathbf{x}}) = \sigma_t g_i(\underbrace{\mathscr{G}_{\sigma}(1, \varsigma * \mathbf{U}_{t+1})\mathbf{Y}_{t+1}}_{\mathscr{G}_{\mathbf{x}}((\mathbf{Y}_t, 1), \varsigma * \mathbf{U}_{t+1}) = \tilde{\mathbf{Y}}_{t+1}} + \bar{\mathbf{x}}) \quad . \tag{38}$$

Therefore,

$$\Gamma_{t+1}^i = \frac{\Gamma_t^i + \frac{\omega_t^i}{d_\gamma} g_i(\tilde{\mathbf{Y}}_{t+1} + \bar{\mathbf{x}})}{\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{t+1})} = \frac{\mathscr{H}_\gamma^{g_i(\cdot+\bar{\mathbf{x}})}(\Gamma_t^i, \omega_t^i, \tilde{\mathbf{Y}}_{t+1})}{\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{t+1})} \ ,$$

for $i = 1, \cdots, m$. Finally,

$$\begin{aligned}
\omega_{t+1}^i &= \mathscr{H}_\omega^{(f,g_i)}(\omega_t^i, \gamma_t^i, \mathbf{X}_t, \mathbf{X}_{t+1}) \\
&= \begin{cases} \omega_t^i \chi^{1/(4d_\omega)} & \text{if } \omega_t^i g_i(\mathbf{X}_{t+1})^2 < k_1 \frac{|h(\mathbf{X}_{t+1}, \gamma_t, \omega_t) - h(\mathbf{X}_t, \gamma_t, \omega_t)|}{n} \\ & \text{or } k_2 |g_i(\mathbf{X}_{t+1}) - g_i(\mathbf{X}_t)| < |g_i(\mathbf{X}_t)| \\ \omega_t^i \chi^{-1/d_\omega} & \text{otherwise, } i = 1, \cdots, m \end{cases} \\
&= \begin{cases} \omega_t^i \chi^{1/(4d_\omega)} & \text{if } \omega_t^i g_i(\tilde{\mathbf{Y}}_{t+1} + \bar{\mathbf{x}})^2 < k_1 \frac{|h(\tilde{\mathbf{Y}}_{t+1} + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma}, \omega_t) - h(\mathbf{Y}_t + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma}, \omega_t)|}{n} \\ & \text{or } k_2 |g_i(\tilde{\mathbf{Y}}_{t+1} + \bar{\mathbf{x}}) - g_i(\mathbf{Y}_t + \bar{\mathbf{x}})| < |g_i(\mathbf{Y}_t + \bar{\mathbf{x}})| \\ \omega_t^i \chi^{-1/d_\omega} & \text{otherwise} \end{cases} \\
&= \mathscr{H}_\omega^{(f(\cdot+\bar{\mathbf{x}}), g_i(\cdot+\bar{\mathbf{x}}))}(\omega_t^i, \Gamma_t^i + \bar{\gamma}^i, \mathbf{Y}_t, \tilde{\mathbf{Y}}_{t+1}) \ ,
\end{aligned}$$

for $i = 1, \cdots, m$, where we used (20), along with (38), and positive homogeneity of $\mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}$ with respect to $[\bar{\mathbf{x}}, \bar{\gamma}]$ to deduce that

$$\begin{aligned}
h(\mathbf{X}_{t+1}, \gamma_t, \omega_t) - h(\mathbf{X}_t, \gamma_t, \omega_t) &= \sigma_t^2 (\mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}(\tilde{\mathbf{Y}}_{t+1} + \bar{\mathbf{x}}, \sigma_t \Gamma_t + \bar{\gamma}) \\
&\quad - \mathscr{D}h_{\bar{\mathbf{x}}, \bar{\gamma}, \omega_t}(\mathbf{Y}_{t+1} + \bar{\mathbf{x}}, \sigma_t \Gamma_t + \bar{\gamma})) \\
&= \sigma_t^2 (h(\tilde{\mathbf{Y}}_{t+1} + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma}, \omega_t) - h(\mathbf{Y}_t + \bar{\mathbf{x}}, \Gamma_t + \bar{\gamma}, \omega_t)) \ .
\end{aligned}$$

$\Phi_{t+1} = (\mathbf{Y}_{t+1}, \Gamma_{t+1}, \omega_{t+1})$ is a function of only $\mathbf{Y}_t$, Γ_t, ω_t, and i.i.d. vectors $\mathbf{U}_{t+1}$. Therefore, $(\Phi_t)_{t \in \mathbb{N}}$ is a homogeneous Markov chain.

A.4 Proof of Corollary 1

By definition, we have

$$h(\mathbf{x}_{\text{opt}} + \alpha \mathbf{x}, \gamma_{\text{opt}} + \alpha \gamma, \omega) = \underbrace{f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{x})}_{A} + \underbrace{\sum_{i=1}^m \gamma^i g_i(\mathbf{x}_{\text{opt}} + \alpha \mathbf{x})}_{B} + \underbrace{\sum_{i=1}^m \frac{\omega^i}{2} g_i(\mathbf{x}_{\text{opt}} + \alpha \mathbf{x})^2}_{C} \ .$$

By developing A, B, and C, we obtain

$$\begin{aligned}
A &= \alpha^2 f(\mathbf{x}_{\text{opt}} + \mathbf{x}) + (1 - \alpha^2) f(\mathbf{x}_{\text{opt}}) + \alpha(1 - \alpha) \underbrace{\mathbf{x}_{\text{opt}}^\intercal \mathbf{H}}_{\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}})} \mathbf{x} \ , \\
B &= \sum_{i=1}^m \alpha^2 (\gamma_{\text{opt}}^i + \gamma^i) g_i(\mathbf{x}_{\text{opt}} + \mathbf{x}) + \alpha(1 - \alpha) \gamma_{\text{opt}}^i \underbrace{\mathrm{b}^i}_{\nabla_{\mathbf{x}} g_i(\mathbf{x}_{\text{opt}})} \mathbf{x} \ , \\
C &= \alpha^2 \sum_{i=1}^m \frac{\omega^i}{2} g_i(\mathbf{x}_{\text{opt}} + \mathbf{x})^2 \ .
\end{aligned}$$

The constraints being active at $\mathbf{x}_{\text{opt}}$, $h(\mathbf{x}_{\text{opt}}, \gamma_{\text{opt}}, \omega) = f(\mathbf{x}_{\text{opt}})$ for all $\omega \in (\mathbb{R}_>^+)^m$. It follows that

$$\mathscr{D}h_{\mathbf{x}_{\text{opt}},\gamma_{\text{opt}},\omega}(\mathbf{x}_{\text{opt}} + \alpha\mathbf{x}, \gamma_{\text{opt}} + \alpha\gamma) = \alpha^2 \Big(f(\mathbf{x}_{\text{opt}} + \mathbf{x}) + \sum_{i=1}^{m} (\gamma_{\text{opt}}^i + \gamma^i) g_i(\mathbf{x}_{\text{opt}} + \mathbf{x}) + \frac{\omega^i}{2} g_i(\mathbf{x}_{\text{opt}} + \mathbf{x})^2 - f(\mathbf{x}_{\text{opt}}) \Big) + \alpha(1-\alpha) \underbrace{\Big(\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{opt}}) + \sum_{i=1}^{m} \nabla_{\mathbf{x}} g_i(\mathbf{x}_{\text{opt}}) \Big)}_{\mathbf{0}} \mathbf{x} \ .$$

The KKT stationarity condition in (29) is satisfied for $\mathbf{x}_{\text{opt}}$ and γ_{opt}. Therefore,

$$\mathscr{D}h_{\mathbf{x}_{\text{opt}},\gamma_{\text{opt}},\omega}(\mathbf{x}_{\text{opt}} + \alpha\mathbf{x}, \gamma_{\text{opt}} + \alpha\gamma) = \alpha^2 \mathscr{D}h_{\mathbf{x}_{\text{opt}},\gamma_{\text{opt}},\omega}(\mathbf{x}_{\text{opt}} + \mathbf{x}, \gamma_{\text{opt}} + \gamma) \ .$$

Consequently, $(\Phi_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain with f convex quadratic.

A.5 Proof of Theorem 3

We express $\frac{1}{t}\ln\frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|}$, $\frac{1}{t}\ln\frac{\|\gamma_t - \gamma_{\text{opt}}\|}{\|\gamma_0 - \gamma_{\text{opt}}\|}$, and $\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0}$ as a function of the homogeneous Markov chain $(\Phi_t)_{t\in\mathbb{N}}$ defined in Theorem 1. Using the property of the logarithm, we have

$$\begin{aligned}\frac{1}{t}\ln\frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|} &= \frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{X}_{k+1} - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_k - \mathbf{x}_{\text{opt}}\|} = \frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{Y}_{k+1}\|}{\|\mathbf{Y}_k\|}\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{k+1}) \\ &= \frac{1}{t}\sum_{k=0}^{t-1}\ln\|\mathbf{Y}_{k+1}\| - \frac{1}{t}\sum_{k=0}^{t-1}\ln\|\mathbf{Y}_k\| + \frac{1}{t}\sum_{k=0}^{t-1}\ln\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{t+1}) \ . \end{aligned} \tag{39}$$

$(\Phi)_{t\in\mathbb{N}}$ is positive Harris-recurrent with an invariant probability measure π and $E_\pi(|\ln\|[\phi]_1\||) < \infty$, $E_\pi(|\ln\|[\phi]_2\||) < \infty$, and $E_\pi(\mathscr{R}(\phi)) < \infty$. Therefore, we can apply Theorem 2 to the right-hand side of (39). We obtain

$$\begin{aligned}\lim_{t\to\infty}\frac{1}{t}\ln\frac{\|\mathbf{X}_t - \mathbf{x}_{\text{opt}}\|}{\|\mathbf{X}_0 - \mathbf{x}_{\text{opt}}\|} &= \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\|\mathbf{Y}_{k+1}\| - \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\|\mathbf{Y}_k\| + \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{t+1}) \\ &= \int \ln\|[\phi]_1\|\pi(d\phi) - \int \ln\|[\phi]_1\|\pi(d\phi) + \int \mathscr{R}(\phi)\pi(d\phi) = -\text{CR} \ . \end{aligned}$$

We proceed similarly with $\frac{1}{t}\ln\frac{\|\gamma_t - \gamma_{\text{opt}}\|}{\|\gamma_0 - \gamma_{\text{opt}}\|}$ and $\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0}$.

$$\frac{1}{t}\ln\frac{\|\gamma_t - \gamma_{\text{opt}}\|}{\|\gamma_0 - \gamma_{\text{opt}}\|} = \frac{1}{t}\sum_{k=0}^{t-1}\ln\|\Gamma_{k+1}\| - \frac{1}{t}\sum_{k=0}^{t-1}\ln\|\Gamma_k\| + \frac{1}{t}\sum_{k=0}^{t-1}\ln\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{t+1}) \ , \tag{40}$$

$$\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0} = \frac{1}{t}\sum_{k=0}^{t-1}\frac{\sigma_{k+1}}{\sigma_k} = \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}\ln\mathscr{G}_\sigma(1, \varsigma * \mathbf{U}_{t+1}) \ . \tag{41}$$

By applying Theorem 2 to the right-hand side of (40) and (41), we obtain

$$\lim_{t\to\infty}\frac{1}{t}\ln\frac{\|\gamma_t - \gamma_{\text{opt}}\|}{\|\gamma_0 - \gamma_{\text{opt}}\|} = \lim_{t\to\infty}\frac{1}{t}\ln\frac{\sigma_t}{\sigma_0} = -\text{CR} \ .$$

Chapter 7

Discussion

We approached two questions related to two different aspects of black-box continuous optimization and adaptive randomized algorithms in this thesis. Our work consists in two parts; in the first one, we tried to address the non-trivial question of how to adapt the step-size efficiently. To that end, we presented a minimal methodology to assess step-size adaptation algorithms in the case of unconstrained optimization. In particular, we presented a realistic test scenario motivated by practical questions that one might consider when designing a new step-size adaptation mechanism.

The second part of our work is the more important and addresses the question of linear convergence of adaptive randomized algorithms for constrained optimization. The context is the following: given an optimization problem with m active linear constraints, we considered an adaptive augmented Lagrangian constraint handling approach that transforms the original constrained problem into a sequence of unconstrained optimization problems of the form

$$\min_{\mathbf{x}} h(\mathbf{x}, \gamma_t, \omega_t) \ ,$$

where h is the augmented Lagrangian, γ_t is the vector of Lagrange factors, and ω_t is the vector of penalty factors, and where γ_t and ω_t are adapted throughout the optimization process. First, we conducted a Markov chain analysis of a $(1+1)$-ES with augmented Lagrangian constraint handling [5] when $m = 1$. This analysis shed light on how linear convergence can be achieved in the more realistic case of $m \geq 1$ linear constraints. In particular, it allowed us to define a practical $(\mu/\mu_W, \lambda)$-ES with augmented Lagrangian constraint handling for $m \geq 1$ constraints, which we analyzed by generalizing the Markov chain analysis conducted previously for $m = 1$. For both studied algorithms, we showed that if the function $\mathscr{D}h_{\bar{\mathbf{x}},\bar{\gamma},\omega}$ defined as

$$\mathscr{D}h_{\bar{\mathbf{x}},\bar{\gamma},\omega}(\mathbf{x}, \gamma) = h(\mathbf{x}, \gamma, \omega) - h(\bar{\mathbf{x}}, \bar{\gamma}, \omega) \ ,$$

is positive homogeneous of degree 2 with respect to $[\bar{\mathbf{x}}, \bar{\gamma}]$ for any $\bar{\gamma}$, ω, and for any $\bar{\mathbf{x}}$ that satisfies $g_i(\bar{\mathbf{x}}) = 0$ for all the constraints g_i, then $(\mathbf{Y}_t, \Gamma_t, \omega_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain, where $\mathbf{Y}_t = \frac{\mathbf{X}_t - \bar{\mathbf{x}}}{\sigma_t}$ is the normalized distance to $\bar{\mathbf{x}}$ and $\Gamma_t = \frac{\gamma_t - \bar{\gamma}}{\sigma_t}$ is the normalized distance to $\bar{\gamma}$. This holds in particular when $\bar{\mathbf{x}}$ is the optimum $\mathbf{x}_{\text{opt}}$ of the problem and $\bar{\gamma}$ is the corresponding vector of Lagrange multipliers γ_{opt}. A key element in constructing the Markov chain was the comparison-based aspect of the studied algorithms and their invariance properties, as well as the updates used for γ_t and ω_t. Linear convergence to $\mathbf{x}_{\text{opt}}$ and γ_{opt} is deduced under stability assumptions on $(\mathbf{Y}_t, \Gamma_t, \omega_t)_{t\in\mathbb{N}}$. These assumptions seem reasonable and were validated empirically by simulations of the Markov chain on the sphere function and a moderately ill-conditioned ellipsoid function. The algorithms we investigated are in fact instances of a more general adaptive augmented Lagrangian approach, where the algorithm iteratively performs one iteration to minimize the augmented Lagrangian, then uses the newly computed solution $\mathbf{X}_{t+1}$ to adapt the parameters γ_t and ω_t of the augmented Lagrangian. Based on this observation, we defined a general framework for building an adaptive randomized algorithm for constrained optimization from another adaptive randomized algorithm for unconstrained optimization, then applied it to a $(\mu/\mu_W, \lambda)$-CMA-ES with median success rule step-size adaptation. This framework was described for one inequality constraint; however, the generalization to m constraints is straightforward.

Our work shows that the Markov chain approach used to analyze linear convergence in the unconstrained case can be extended to the constrained case when considering linear constraints and an augmented Lagrangian approach for handling them. As mentioned above, we only assume the stability of the Markov chain to deduce linear convergence. Therefore, an interesting extension for our analysis would be to prove the stability (positivity and Harris-recurrence) of the chain $(\mathbf{Y}_t, \Gamma_t, \omega_t)_{t\in\mathbb{N}}$ in order to have a complete proof of convergence. Another possible extension would be to consider *general* update rules for γ_t and ω_t in the analysis and give sufficient conditions on these update rules such that a homogeneous Markov chain with the desired stability properties exists. Finally, the general $(\mu/\mu_W, \lambda)$-ES with adaptive augmented Lagrangian we present for handling m constraints can be seen as a prototype. Indeed, although the preliminary results are interesting, in that linear convergence can be achieved on a class of problems when the constraints are linear, a more thorough empirical validation is required. In particular, the algorithm needs to be tested on different test scenarios, in higher dimensions, and with a larger number of constraints.

References

[1] O. Ait Elhara, A. Auger, and N. Hansen. A Median Success Rule for Non-Elitist Evolution Strategies: Study of Feasibility. In *Genetic and Evolutionary Computation Conference*, pages 415–422. ACM Press, 2013.

[2] D. V. Arnold. On the Behaviour of the $(1, \lambda)$-σSA-ES for a Constrained Linear Problem. In C. A. Coello Coello et al., editors, *Parallel Problem Solving from Nature, PPSN XII*, pages 82–91. Springer, 2012.

[3] D. V. Arnold. An Active-Set Evolution Strategy for Optimization with Known Constraints. In *Parallel Problem Solving from Nature, PPSN XIV*, pages 192–202. Springer, 2016.

[4] D. V. Arnold and N. Hansen. A $(1+1)$-CMA-ES for Constrained Optimisation. In *Genetic and Evolutionary Computation Conference*, pages 297–304. ACM Press, 2012.

[5] D. V. Arnold and J. Porter. Towards an Augmented Lagrangian Constraint Handling Approach for the $(1+1)$-ES. In *Genetic and Evolutionary Computation Conference*, pages 249–256. ACM Press, 2015.

[6] A. Atamna. Benchmarking IPOP-CMA-ES-TPA and IPOP-CMA-ES-MSR on the BBOB Noiseless Testbed. In *Genetic and Evolutionary Computation Conference Companion*, pages 1135–1142. ACM Press, 2015.

[7] A. Atamna, A. Auger, and N. Hansen. Analysis of Linear Convergence of a $(1+1)$-ES with Augmented Lagrangian Constraint Handling. In *Genetic and Evolutionary Computation Conference*, pages 213–220. ACM Press, 2016.

[8] A. Atamna, A. Auger, and N. Hansen. Augmented Lagrangian Constraint Handling for CMA-ES—Case of a Single Linear Constraint. In *Parallel Problem Solving from Nature, PPSN XIV*, pages 181–191. Springer, 2016.

[9] C. Audet and J. E. Dennis Jr. Analysis of Generalized Pattern Searches. *SIAM Journal on Optimization*, 13(3):889–903, 2003.

[10] A. Auger. Convergence Results for the $(1, \lambda)$-SA-ES Using the Theory of φ-Irreducible Markov Chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[11] A. Auger. *Analysis of Comparison-Based Stochastic Continuous Black-Box Optimization Algorithms*. "Habilitation à diriger les recherches" dissertation, Université Paris-Sud, 2015.

[12] A. Auger and N. Hansen. Performance Evaluation of an Advanced Local Search Evolutionary Algorithm. In *IEEE Congress on Evolutionary Computation*, pages 1777–1784. IEEE, 2005.

[13] A. Auger and N. Hansen. Reconsidering the Progress Rate Theory for Evolution Strategies in Finite Dimensions. In *Genetic and Evolutionary Computation Conference*, pages 445–452. ACM Press, 2006.

[14] A. Auger and N. Hansen. Linear Convergence on Positively Homogeneous Functions of a Comparison Based Step-Size Adaptive Randomized Search: the $(1+1)$-ES with Generalized One-Fifth Success Rule. Submitted for publication, 2013.

[15] A. Auger and N. Hansen. Linear Convergence of Comparison-Based Step-Size Adaptive Randomized Search via Stability of Markov Chains. *SIAM Journal on Optimization*, 26(3):1589–1624, 2016.

[16] A. Auger, N. Hansen, J. M. Perez Zerpa, and M. Schoenauer. Experimental Comparison of Derivative Free Optimization Algorithms. In 8*th International Symposium on Experimental Algorithms*, volume 5526 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2009.

[17] H. J. C. Barbosa, A. C. C. Lemonge, and H. S. Bernardino. A Critical Review of Adaptive Penalty Techniques in Evolutionary Computation. In R. Datta and K. Deb, editors, *Evolutionary Constrained Optimization*, chapter 1, pages 1–27. Springer, 2015.

[18] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[19] H.-G. Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer-Verlag, 2001.

[20] H.-G. Beyer and H.-P. Schwefel. Evolution Strategies: A Comprehensive Introduction. *Natural Computing*, 1(1):3–52, 2002.

[21] A. Bienvenüe and O. François. Global Convergence of Evolution Strategies in Spherical Problems: Some Simple Proofs and Difficulties. *Theoretical Computer Science*, 306(1–3):269–289, 2003.

[22] E. G. Birgin, C. A. Floudas, and J. M. Martínez. Global Minimization Using an Augmented Lagrangian Method with Variable Lower-Level Constraints. *Mathematical Programming*, 125(1):139–162, 2010.

[23] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. SIAM, 2014.

[24] P. T. Boggs and J. W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4(4):1–52, 1995.

[25] A. Chotard and A. Auger. Verifiable Conditions for Irreducibility, Aperiodicity and T-chain Property of a General Markov Chain. Submitted for publication, 2015.

[26] A. Chotard, A. Auger, and N. Hansen. Cumulative Step-Size Adaptation on Linear Functions. In *Parallel Problem Solving from Nature, PPSN XII*, pages 72–81. Springer, 2012.

[27] A. Chotard, A. Auger, and N. Hansen. Markov Chain Analysis of Cumulative Step-Size Adaptation on a Linear Constraint Problem. *Evolutionary Computation*, 23(4):611–640, 2015.

[28] A. Chotard and M. Holeňa. A Generalized Markov-Chain Modelling Approach to $(1,\lambda)$-ES Linear Optimization. In *Parallel Problem Solving from Nature, PPSN XIII*, pages 902–911. Springer International Publishing, 2014.

[29] M. Clerc and J. Kennedy. The Particle Swarm—Explosion, Stability, and Convergence in Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[30] C. A. Coello Coello. Constraint-Handling Techniques Used with Evolutionary Algorithms. In *Proceedings of the* 14*th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 849–871. ACM, 2012.

[31] 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*, chapter 12, pages 487–496. Wiley, 2010.

[32] A. R. Conn, N. I. M. Gould, and P. L. Toint. A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.

[33] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MOS-SIAM Series on Optimization. SIAM, 2000.

[34] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization. SIAM, 2009.

[35] W. C. Davidon. Variable Metric Method for Minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.

[36] K. Deb and S. Srivastava. A Genetic Algorithm Based Augmented Lagrangian Method for Constrained Optimization. *Computational Optimization and Applications*, 53(3):869–902, 2012.

[37] J. E. Dennis and J. J. Moré. Quasi-Newton Methods, Motivation and Theory. *SIAM Review*, 19(1):46–89, 1977.

[38] J. E. Dennis Jr. and V. Torczon. Derivative-Free Pattern Search Methods for Multidisciplinary Design Problems. In *Proceedings of the* 5*th Symposium on Multidisciplinary Analysis and Optimization*, 1994.

[39] P. Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, volume 35. Springer, 2011.

[40] G. Di Pillo and L. Grippo. An Exact Penalty Funtion Method with Global Convergence Properties for Nonlinear Programming. *Mathematical Programming*, 36(1):1–18, 1986.

[41] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag, 2nd edition, 2003.

[42] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM, 1990.

[43] R. Fletcher. An Exact Penalty Function for Nonlinear Programming with Inequalities. *Mathematical Programming*, 5:129–150, 1973.

[44] R. Fletcher. *Practical Methods of Optimization*. Wiley-Interscience, 2nd edition, 1987.

[45] C. J. Geyer. Markov Chain Monte Carlo Lecture Notes. http://www.stat.umn.edu/geyer/f05/8931/n1998.pdf, 1998.

[46] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[47] S.-P. Han and O. L. Mangasarian Mangasarian. Exact Penalty Funtions in Nonlinear Programming. *Mathematical Programming*, 17(1):251–269, 1979.

[48] H. Hansen, D. V. Arnold, and A. Auger. Evolution strategies. In J. Kacprzyk and W. Pedrycz, editors, *Handbook of Computational Intelligence*, chapter 44, pages 871–898. Springer, 2015.

[49] N. Hansen. CMA-ES with Two-Point Step-Size Adaptation. Research report, Inria, 2008.

[50] N. Hansen. The CMA Evolution Strategy: A Tutorial. http://arxiv.org/pdf/1604.00772v1.pdf, 2016.

[51] N. Hansen, A. Atamna, and A. Auger. How to Assess Step-Size Adaptation Mechanisms in Randomized Search. In *Parallel Problem Solving from Nature, PPSN XIII*, pages 60–69. Springer, 2014.

[52] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. http://numbbo.github.io/coco-doc/, 2016.

[53] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[54] 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.

[55] A. L. Haupt and S. E. Haupt. *Practical Genetic Algorithms*. Wiley, 2nd edition, 2004.

[56] M. R. Hestenes. Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[57] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1992.

[58] R. Hooke and T. A. Jeeves. "Direct Search" Solution of Numerical and Statistical Problems. *Journal of the ACM*, 8(2):212–229, 1961.

[59] N. Karmarkar. A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, 4(4):373–395, 1984.

[60] J. Kennedy and R. C. Eberhart. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks*. IEEE, 1995.

[61] S. Koziel and Z. Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[62] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.

[63] J. Lampinen and I. Zelinka. On Stagnation of the Differential Evolution Algorithm. In *International Conference on Soft Computing MENDEL*, pages 76–83, 2000.

[64] R. Le Riche and F. Guyon. Dual Evolutionary Optimization. In *Artificial Evolution*, volume 2310 of *Lecture Notes in Compter Science*, pages 281–294. Springer-Verlag, 2002.

[65] R. M. Lewis and V. Torczon. A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds. *SIAM Journal on Optimization*, 12(4):1075–1089, 2002.

[66] K. I. M. McKinnon. Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

[67] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, 1993.

[68] E. Mezura-Montes and C. A. Coello Coello. Constrained Optimization via Multiobjective Evolutionary Algorithms. In *Multiobjective Problem Solving from Nature: From Concepts to Applications*, Natural Computing Series, pages 53–75. Springer, 2008.

[69] 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.

[70] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996.

[71] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[72] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 1965.

[73] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[74] R. Poli, J. Kennedy, and T. Blackwell. Particle Swarm Optimization: An Overview. *Swarm Intelligence*, 1(1):33–57, 2007.

[75] F. A. Potra and S. J. Wright. Interior-Point Methods. *Journal of Computational and Applied Mathematics*, 124:281–302, 2000.

[76] M. J. D. Powell. A Method for Nonlinear Constraints in Minimization Problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, 1969.

[77] M. J. D. Powell. The NEWUOA Software for Unconstrained Optimization without Derivatives. In *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and its Applications*, pages 255–297. Springer, 2006.

[78] M. J. D. Powell. Developments of NEWUOA for Minimization without Derivatives. *IMA Journal of Numerical Analysis*, 28(4):649–664, 2008.

[79] M. J. D. Powell. The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives. Technical report, Department of Applied Mathematics and Theoretical Physics, Cambridge University, 2009.

[80] K. V. Price. Differential Evolution vs. the Functions of the 2nd ICEO. In *IEEE International Conference on Evolutionary Computation*, pages 153–157, 1997.

[81] I. Rechenberg. Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. *Problemata*, 1973.

[82] R. Salomon. Evolutionary Algorithms and Gradient Search: Similarities and Differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.

[83] Y. Shi and R. C. Eberhart. Empirical Study of Particle Swarm Optimization. In *Congress on Evolutionary Computation*, volume 3, pages 1945–1949, 1999.

[84] A. Smith and D. W. Coit. Constraint-Handling Techniques—Penalty Functions. In *Handbook of Evolutionary Computation*, chapter C5.2. Institute of Physics Publishing and Oxford University Press, 1997.

[85] F. Solis and R. J.-B. Wets. Minimization by Random Search Techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.

[86] R. Storn and K. Price. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[87] M.-J. Tahk and B.-C. Sun. Coevolutionary Augmented Lagrangian Methods for Constrained Optimization. *IEE Transactions on Evolutionary Computation*, 4(2):114–124, 2000.

[88] V. Torczon. On the Convergence of Pattern Search Algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.

[89] A. H. Wright. Genetic Algorithms for Real Parameter Optimization. In *Foundations of Genetic Algorithms*, pages 205–218. Morgan Kaufmann, 1991.

[90] X. Yu and M. Gen. *Introduction to Evolutionary Algorithms*. Springer-Verlag, 2010.

[91] Y. Yuan. A Review of Trust Region Algorithms for Optimization. In 4*th International Congress on Industrial and Applied Mathematics*, volume 99, pages 271–282. Oxford University Press, 2000.

[92] Y. Yuan. Recent Advances in Trust Region Algorithms. *Mathematical Programming*, 151(1):249–281, 2015.

[93] A. Zhigljavsky and A. Žilinskas. *Stochastic Global Optimization*. Springer Optimization and Its Applications. Springer, 2008.