[EURO Journal on Computational Optimization 9 (2021) 100001](https://doi.org/10.1016/j.ejco.2020.100001)

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

EURO Journal on Computational Optimization

journal homepage: [www.elsevier.com/locate/ejco](http://www.elsevier.com/locate/ejco)

A merit function approach for evolution strategies

Youssef Diouane[1](#_bookmark1)

*ISAE-SUPAERO, Université de Toulouse, Toulouse Cedex 4 31055, France*

a r t i c l e i n f o a b s t r a c t

*Keywords:*

Constrained optimization Derivative-free optimization Evolution strategy

Merit function Global convergence

In this paper, we extend a class of globally convergent evolution strategies to handle general constrained opti- mization problems. The proposed framework handles quantifiable relaxable constraints using a merit function approach combined with a specific restoration procedure. The unrelaxable constraints, when present, can be treated either by using the extreme barrier function or through a projection approach. Under reasonable as- sumptions, the introduced extension guarantees to the regarded class of evolution strategies global convergence properties for first order stationary constraints. Numerical experiments are carried out on a set of problems from the CUTEst collection as well as on known global optimization problems.

# Introduction

In this paper, we are interested in constrained derivative-free opti- mization problems ([Audet and Hare, 2017](#_bookmark45)), i.e.,

min *𝑓* (*𝑥*)

In [Diouane et al. (2015b)](#_bookmark27), the authors proposed a general glob- ally convergent framework for unrelaxable constraints using two differ- ent approaches. The first relies on techniques inspired from directional direct-search methods ([Conn et al., 2009; Kolda et al., 2003](#_bookmark24)), where one uses an extreme barrier function to prevent unfeasible displacements

s.t. *𝑥* ∈ Ω = Ω

∩Ω *,*

*𝑞𝑟*

(1)

*𝑢𝑟*

together with the possible use of directions that conform to the local geometry of the feasible region. The second approach was based on en-

ous. The feasible region Ω *⊂* ℝ*𝑛* of this problem includes two categories where the objective function *f* is assumed to be locally Lipschitz continu- of constraints ([Le Digabel and Wild, 2015](#_bookmark47)). The first, denoted by Ω*qr* and

known as quantifiable relaxable (QR) constraints, or soft constraints, is allowed to be violated during the optimization process and may need to be satisfied only approximately or asymptotically. Such a set of con- straints will be assumed, in the context of this paper, to be of the form:

Ω*𝑞𝑟* = {*𝑥* ∈ ℝ*𝑛* ∀*𝑖* ∈ {1*,* … *, 𝑟*}*, 𝑐𝑖*(*𝑥*) ≤ 0}*,*

|

egory of constraints, denoted by Ω*𝑢𝑟 ⊂* ℝ*𝑛,* pools all unrelaxable (UR) where the functions *ci* are locally Lipschitz continuous. The second cat-

constraints (also known as hard constraints), for such constraints no violation is allowed and they require satisfaction during the entire op- timization process.

Evolution strategies (ES’s) ([Rechenberg, 1973](#_bookmark56)) are evolutionary al- gorithms designed for global optimization in a continuous space, and [that lead to promising results on practical optimization problems (Auger](#_bookmark53) [et](#_bookmark26) [al., 2009; Bouzarkouna, 2012; Rios and Sahinidis, 2010). In Diouane](#_bookmark53)

certain number *𝜆* of points (called offspring) are randomly generated in [et al. (2015a,b), the authors dealt with a large class of ES’s, where a](#_bookmark26) each iteration, among which *𝜇* ≤ *𝜆* of them (called parents) are selected.

ES’s have been growing rapidly in popularity and used for solving chal- lenging optimization problems ([Auger et al., 2013; Hansen et al., 2010](#_bookmark51)).

*E-mail address:* [youssef.diouane@isae-supaero.fr](mailto:youssef.diouane@isae-supaero.fr)

1 <https://orcid.org/0000-0002-6609-7330>

forcing all the generated sample points to be feasible, while using a

projection mapping approach. Both proposed strategies were compared to some of the best available solvers for minimizing a function with- out derivatives. The numerical results confirmed the competitiveness of the two approaches in terms of eﬃciency as well as robustness. Moti- vated by the recent availability of massively parallel computing plat- forms, the authors in [Diouane et al. (2016)](#_bookmark28) proposed a highly parallel globally convergent ES (inspired by [Diouane et al. (2015b)](#_bookmark27)) adapted to the full-waveform inversion setting. By combining model reduction and ES’s in a parallel environment, the authors contributed solving realistic instances of the full-waveform inversion problem.

In the context of ES’s, many algorithms have been proposed in the literature to adapt ES’s to solve constrained optimization problems ([Coello, 0000](#_bookmark58)). [Coello (2002)](#_bookmark59) and [Kramer (2010)](#_bookmark42) out- lined a comprehensive survey of the most popular constraints han- dling methods currently used with ES’s. Recently, the authors in [Atamna et al. (2018)](#_bookmark40) proposed an adaptation of a class of ES’s to han- dle QR constraints by using an augmented Lagrangian framework. The proposed approach was showed to enjoy good local and invariant con- vergence properties. To the best of our knowledge, all the ES’s proposed suffer from the lack of global convergence guarantees when applied to general constrained optimization problems.

In the context of deterministic derivative-free optimization (DFO), only few works looked at both kinds (relaxable and unrelaxable) of con-

<https://doi.org/10.1016/j.ejco.2020.100001> Received 24 August 2020; Accepted 25 August 2020

2192-4406/© 2020 The Author. Published by Elsevier Ltd on behalf of Association of European Operational Research Societies (EURO). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

straints separately. For instance, [Audet and Dennis Jr. (2009)](#_bookmark49) outlined a globally convergent direct-search approach based on a progressive bar- rier, which combined an extreme barrier approach for unrelaxable con- straints and non-dominance filters ([Fletcher and Leyffer, 2002](#_bookmark31)) to han- dle QR constraints. More recently, the authors in [Audet et al. (2018)](#_bookmark43) ex- [tended the progressive barrier approach, developed in Audet and Den- nis Jr. (2009), to cover the setting of a derivative-free trust-region](#_bookmark49) method. Within the framework of directional direct-search methods, [Gratton and Vicente (2014)](#_bookmark33) proposed an alternative where one handles QR constraints by means of a merit function. Under the appropriate assumptions, the latter approach ensured global convergence by impos- ing a suﬃcient decrease condition on a merit function combining infor- mation from both objective function and constraint violation. Another

size parameter is also possible. The latter increases or decreases depend- ing on the landscape of the objective function. One relevant instance of [such an ES is covariance matrix adaptation ES (CMA-ES) (Hansen et al., 1995).](#_bookmark34)

In [Diouane et al. (2015a,b)](#_bookmark26), the authors proposed a framework for making a class of ES’s enjoying some global convergence properties while solving optimization problems possibly with UR constraints. In fact, in [Diouane et al. (2015a)](#_bookmark26), by imposing a suﬃcient decreasing con- dition on the objective function value, the proposed algorithm moni-

tored the step size *𝜎k* to ensure its convergence to zero (which leads

creasing condition is applied directly to the weighted mean *𝑥𝑡𝑟𝑖𝑎𝑙* of then to the existence of a stationary point). The imposed suﬃcient de-

*𝑘*+1

the new parents. By suﬃcient decreasing condition we mean *𝑓* (*𝑥𝑡𝑟𝑖𝑎𝑙* ) ≤

[two-phases derivative-free approach was proposed in Martínez and So-](#_bookmark50)

*𝑓* (*𝑥* ) − *𝜌*(*𝜎* )*,* where *𝜌*

*𝑘*+1

*𝑘 𝑘*

( ·) is a forcing function ([Kolda et al., 2003](#_bookmark41)), i.e., a

[bral (2013) to specifically handle the case where finding a feasible point](#_bookmark50) is easier than minimizing the objective function.

In this paper, inspired by the merit function approach for direct search methods ([Gratton and Vicente, 2014](#_bookmark33)), we propose to adapt a class of ES algorithms (as proposed in [Diouane et al. (2015b)](#_bookmark27)) to handle both QR and unrelaxable constraints. The class of ES algorithms ob- tained relies essentially on a merit function (eventually with a restora- tion procedure) to decide and control the distribution of the offspring

points. The merit function is a standard penalty-based function that has

positive, nondecreasing function satisfying *𝜌*(*𝜎*)/*𝜎* → 0 when *𝜎* → 0. To

handle UR constraints ([Diouane et al., 2015b](#_bookmark27)), one starts with a feasible

iterate *x*0 and then aviods stepping outside the feasible region by means of a barrier approach. In this context, the suﬃcient decrease condition is

applied not to *f* but to the extreme barrier function *𝑓*Ω*𝑢𝑟* associated with *f*

with respect to the constraints set Ω*ur* ([Audet and Dennis Jr., 2006](#_bookmark48)) (also

known as the death penalty function in the terminology of evolutionary

algorithms), which is defined by:

already been proposed in the context of ES ([Coello, 2002](#_bookmark59)). The main advantage of the proposed approach is to ensure a form of global con-

*𝑓*Ω*𝑢𝑟*

*𝑓* (*𝑥*) if *𝑥* ∈ Ω*𝑢𝑟,*

+∞ otherwise.

(*𝑥*) = {

vergence. Namely, under reasonable assumptions, this paper presents the first globally convergent ES framework handling both QR and UR constraints.

The proposed convergence theory generalizes the ES framework in [Diouane et al. (2015b)](#_bookmark27) by including QR constraints, all in the spirit of the proposed merit function for directional direct search meth- ods ([Gratton and Vicente, 2014](#_bookmark33)). The contribution of this paper is twofold. First, we propose an adaptation of the merit function approach algorithm to the ES setting, a detailed convergence theory of the pro- posed approach is given. Second, we provide a practical implementation and extensive tests on a set of problems from the CUTEst collection as well as on known global optimization problems. The performance of our proposed solver is compared to (a) the progressive barrier approach im- plemented in the NOMAD solver ([Le Digabel, 2011](#_bookmark46)), (b) the directional direct search method as proposed in [Gratton and Vicente (2014)](#_bookmark33) and

(c) an adaptation of a well known ES using an augmented Lagrangian approach to handle QR constraints ([Atamna et al., 2018](#_bookmark40)).

The paper is organized as follows. The proposed merit function ap- proach is given in [Section 2](#_bookmark2) with a detailed description of the changes introduced in a class of ES algorithms in order to handle general con- straints. The convergence results of the adapted approach are then de- tailed in [Section 3](#_bookmark4). In [Section 4](#_bookmark16), we test the proposed algorithm on a set of problems from the CUTEst collection as well as on known global

[The extreme barrier function is formally introduced in Audet and](#_bookmark45) [Hare](#_bookmark27) [(2017). The obtained ES approach is detailed in (Diouane et al.,](#_bookmark45) [2015b, Algorithm 2.1). The global convergence of the algorithm is](#_bookmark27) achieved by establishing that some type of directional derivatives are nonnegative at limit points of refining subsequences along certain limit directions (see [Diouane et al., 2015b](#_bookmark27), Theorem 2.1).

[The challenge of this paper consists in extending (Diouane et al., 2015b, Algorithm 2.1) to a globally convergent framework that takes](#_bookmark27) into account both QR and UR constraints. The author acknowledges that a preliminary version of this work was produced during his PhD thesis [(Diouane, 2014, Chapter 5)](#_bookmark29). In what comes next, we define the merit function as follows:

*𝑓* (*𝑥*) + *𝛿̄𝑔*(*𝑥*) if *𝑥* ∈ Ω*𝑢𝑟,*

*𝑀* (*𝑥*) = {

+∞ otherwise.

where *𝛿̄ >* 0 is a given positive constant and *g* defines a constraint viola-

tion function with respect to QR constraints. The 𝓁1-norm is commonly

used to define the constraint violation function, i.e.,

*𝑟*

*𝑔*(*𝑥*) = max{*𝑐𝑖* (*𝑥*)*,* 0}*.*

∑

*𝑖*=1

Other choices for *g* exist, for instance, using the 𝓁2-norm i.e., *𝑔*(*𝑥*) =

∑

optimization problems. Finally, we make some concluding remarks in

*𝑟*

*𝑖*=1

max{*𝑐𝑖* (*𝑥*)*,* 0}2 . We note that the same constraint violation function

[Section 5](#_bookmark32).

# A globally convergent ES for general constraints

This paper focuses on a class of ES’s, denoted by (*𝜇*/*𝜇W, 𝜆*)-ES, which evolves a single candidate solution. In fact, at the *𝑘*−th iteration, a new population *𝑦*1 *,* … *, 𝑦𝜆* (called offspring) is generated around a

*𝑘*+1

*𝑘*+1

bol “/*𝜇W* ” in (*𝜇*/*𝜇W, 𝜆*)-ES specifies that *𝜇* parents are “recombined” into weighted mean *xk* of the previous parents (candidate solution). The sym- a weighted mean. The parents are selected as the *𝜇* best offspring of the

previous iteration in terms of the objective function value. The muta-

[*g* is used within the progressive barrier approach (Audet and Dennis Jr.,](#_bookmark49)

[2009](#_bookmark31)[), that was in turn inspired by the filter approach of Fletcher and](#_bookmark49) [Leyffer (2002). The merit function will be used to evaluate a trial step](#_bookmark31) and hence decide whether such step will be accepted or not. The ex- tension of the globally convergent ES to a general constrained setting can be seen as a combination of two approaches, a feasible one where either the extreme barrier or a projection operator will be used to han- dle the UR constraints, and a merit function approach (possibly with a restoration procedure) to handle QR constraints.

iteration *k*, a trial mean parent *𝑥𝑡𝑟𝑖𝑎𝑙* is computed as the weighted mean The description of the proposed framework is as follows. For a given

*𝑘*+1

tion operator of the new offspring points is done by *𝑦𝑖*

= *𝑥𝑘* + *𝜎𝐸𝑆 𝑑𝑖 ,*

of the *𝜇* best points in terms of the merit function value. The current

*𝑘*+1

*𝑘 𝑘*

*𝑖* = 1*,* … *, 𝜆,* where *𝑑𝑖* is drawn from a certain distribution C*𝑘* and *𝜎𝐸𝑆*

trial mean parent will be considered as a “***Successful point***” if one of

*𝑘 𝑘*

long to the simplex set *𝑆* = {(*𝜔*1*,* … *, 𝜔𝜇*) ∈ ℝ*𝜇* ∶ ∑*𝜇 𝑤𝑖* = 1*, 𝑤𝑖* ≥ 0*, 𝑖* = is a chosen step size. The weights used to compute the means be-

*𝑖*=1

is suﬃciently away from the feasible region (i.e., *g*(*xk* ) *> C𝜌*(*𝜎k* ) for the two following situations occur. The first scenario arises when one

*W* )-ES adapts the sampling distribution to the land- scape of the objective function. An adaptation mechanism for the step

1*,* … *, 𝜇*}. The (*𝜇*/*𝜇*

*, 𝜆*

lation function *g* (i.e., *𝑔*Ω

*𝑘*+1

(*𝑥𝑡𝑟𝑖𝑎𝑙*) *< 𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* )*,* where *𝑔*Ω

denotes the

*𝑢𝑟*

some constant *C >* 1) and *𝑥𝑡𝑟𝑖𝑎𝑙* suﬃciently decreases the constraint vio-

*𝑘*+1

*𝑢𝑟*

extreme barrier function associated with *g* with respect to Ω*ur*). The sec-

(i.e., *𝑀* (*𝑥𝑡𝑟𝑖𝑎𝑙*) *< 𝑀* (*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* )). ond situation occurs when the merit function is suﬃciently decreased

*𝑘*+1

Before checking whether the trial point is successful or not, the al- gorithm will try first to restore the feasibility or at least decrease the

**Algorithm 1:** A globally convergent ES for general constraints (Main).

**Data**: choose positive integers *𝜆* and *𝜇* such that *𝜆* ≥ *𝜇*.Select an initial *𝑥*0 ∈ Ω*𝑢𝑟* and evaluate *𝑓* (*𝑥*0 ).Choose initial step

lengths *𝜎*0 *, 𝜎𝐸𝑆 >* 0 and initialweights (*𝜔*1*,* … *, 𝜔𝜇*) ∈ *𝑆*.

constraint violation if needed. The restoration process will be activated 0 0 0

trial point *𝑥𝑡𝑟𝑖𝑎𝑙* suﬃciently decreases the constraint violation function *g* if the current mean parent *xk* is far away from the feasible region and the

*𝑘*+1

but not the merit function. More specifically, a “***Restoration identiﬁer***” will be activated if one has

*𝑔*Ω (*𝑥𝑡𝑟𝑖𝑎𝑙* ) *< 𝑔*(*𝑥* ) − *𝜌*(*𝜎* ) and *𝑔*(*𝑥* ) *> 𝐶𝜌*(*𝜎* )

Choose constants *𝛽*1 *, 𝛽*2 *, 𝑑*min *, 𝑑*max suchthat 0 *< 𝛽*1 ≤ *𝛽*2 *<* 1

and 0 *< 𝑑*min *< 𝑑*max. Select a forcing function *𝜌*(⋅)

**1 for** *𝑘* = 0*,* 1*,* … **do**

**2 Step 1:** compute new sample points

*𝑌𝑘*+1 = {*𝑦*1 *,* … *, 𝑦𝜆* }such that

*𝑘*+1

*𝑘*+1

*𝑢𝑟*

and

*𝑘*+1

*𝑘 𝑘*

*𝑘 𝑘*

*𝑖*

*𝑘*+1

*𝑦*

= *𝑥𝑘*

+ *𝜎𝑘 𝑑̃𝑖 , 𝑖* = 1*,* … *, 𝜆,*

*𝑀* (*𝑥𝑡𝑟𝑖𝑎𝑙* ) ≥ *𝑀* (*𝑥𝑘* )*.*

*𝑘*

*𝑘*+1

The restoration algorithm will be left as far as progress on the re-

**3**

**Step 2:** evaluate *𝑀* (*𝑦𝑖*

where the directions *𝑑̃𝑖* ’s are computed from the original ES

directions *𝑑𝑖* ’s(which in turn are drawn from a chosen ES

*𝑘*

*𝑘*

distribution C*𝑘* and scaled if necessary to

*𝑑𝑖*

≤

duction of the constraint violation can not be achieved all without any considerable increase in *f*. The complete description of the restoration

satisfy*𝑑*

min

‖ *𝑘*‖

≤ *𝑑*

max

).;

procedure is given in [Algorithm 2](#_bookmark6).

As a result, the main iteration of the proposed merit function ap- proach can be divided into two steps: restoration and minimization. In

essentially the function *𝑔*Ω*𝑢𝑟* ) while in the minimization step the objec- the restoration step the aim is to decrease infeasibility (by minimizing

tive function *f* is improved over a relaxed set of constraints by using the

*𝑘*+1

points in *𝑌𝑘*+1 = {*𝑦̃𝑘*+1 *,* … *, 𝑦̃𝑘*+1 }by increasing order:

1 *𝜆*

*𝑀* (*𝑦̃*1 ) ≤ ⋯ ≤ *𝑀* (*𝑦̃𝜆* ).Select the new parents as the best *𝜇*

offspring sample points{*𝑦̃*1 *,* … *, 𝑦̃𝜇* }, and compute their

*𝑘*+1

*𝑘*+1

*𝑘*+1

*𝑘*+1

weighted mean

*𝜇*

= ∑ *𝜔 𝑦̃*

), *𝑖* = 1*,* … *, 𝜆*, and reorder theoffspring

;

merit function *M*. The final approach obtained is described is given in

[Algorithm 1](#_bookmark3).

*𝑡𝑟𝑖𝑎𝑙*

*𝑘*+1

*𝑥*

*𝑖 𝑖*

*𝑘 𝑘*+1

*𝑖*=1

For both algorithms (main and restoration), our global convergence

analysis will be performed independently of the choice of the distribu-

tion C*𝑘,* the weights (*𝜔*1 *,* … *, 𝜔𝜇*) ∈ *𝑆,* and the step size *𝜎𝐸𝑆* . Therefore,

**Step 3: if** *𝑥𝑡𝑟𝑖𝑎𝑙* ∉ Ω*𝑢𝑟* **then**

1. the iteration is declared unsuccessful;

*𝑘*+1

# else

*𝑘 𝑘 𝑘*

the update of the ES parameters is left unspecified at this stage. How- **6**

ever, the distribution C*𝑘* will be very useful in ensuring that a central **7**

**if** *𝑥𝑡𝑟𝑖𝑎𝑙 is a “****Restoration identiﬁer****”* **then**

enter Restoration (with *𝑘𝑟* = *𝑘*);

*𝑘*+1

convergence assumption (related to the density of the directions in the unit sphere) can be seen as reasonable. In fact, by choosing the distribu-

tion C*𝑘* to be multivariate normal distribution with mean zero, one can

guarantee the density of the directions with a probability one. We will

give more details on that in the next section.

Note that we also impose bounds on all directions *𝑑𝑖* used by the al-

*𝑘*

gorithm. This modification is, however, very mild since the lower bound

a very large number. The construction of the set of directions {*𝑑̃𝑖* } can *d*min can be chosen very close to zero and the upper bound *d*max set to

*𝑘*

be done with respect to the local geometry of the UR constraints as pro- posed in [(Diouane et al., 2015b, Section 2.2)](#_bookmark27).

# Global convergence

The convergence results presented in this section are in the vein of those first established for the merit function approach for direct search methods ([Gratton and Vicente, 2014](#_bookmark33)). For the convergence analysis, we will consider a sequence of iterations generated by [Algorithm 1](#_bookmark3) with- out any stopping criterion. The analysis is organized depending on the number of times restoration is entered.

* 1. *Case 1: the restoration algorithm is never entered after a certain order*

a subsequence of the step sizes {*𝜎k* } will converge to zero. In fact, due to When the restoration is entered finite times, one can guarantee that

the suﬃcient decrease condition imposed on the merit function along the iterates (or in the constraints violation function if the iterates are

size (reduced at least by *𝛽*2 for unsuccessful iterations), one can ensure suﬃciently away from the feasible region) and the control on the step

the existence of a subsequence *K* of unsuccessful iterates driving the step size to zero.

**Lemma 3.1.** *Let f be bounded below and assuming that the restoration is not entered after a certain order. Then,*

lim inf *𝜎* = 0*.*

*𝑘*→+∞

*𝑘*

# else

* + 1. **if** *𝑥𝑡𝑟𝑖𝑎𝑙 is a “****Successful point****”* **then**

*𝑘*+1

* + 1. declare the iteration successful, set *𝑥𝑘*+1 = *𝑥𝑡𝑟𝑖𝑎𝑙* , and*𝜎𝑘*+1 ≥ *𝜎𝑘* (for example *𝜎𝑘*+1 = max{*𝜎𝑘 , 𝜎𝐸𝑆* });

*𝑘*

*𝑘*+1

# else

* + 1. the iteration is declared unsuccessful;
    2. **end**
    3. **end**
    4. **end**
    5. **if** *the iteration is declared unsuccessful* **then**

**17** set *𝑥𝑘*+1 = *𝑥𝑘* and*𝜎𝑘*+1 = *𝛽𝑘 𝜎𝑘* , with *𝛽𝑘* ∈ (*𝛽*1 *, 𝛽*2 );

# 18 end

**19 Step 4:** update the ES step length *𝜎𝐸𝑆* , the distribution C*𝑘*+1 , and the weights(*𝜔*1 *,* … *, 𝜔𝜇* ) ∈ *𝑆*;

*𝑘*+1

*𝑘*+1

*𝑘*+1

# 20 end

**Proof.** Suppose that there exists a *𝑘̄ >* 0 and *𝜎 >* 0 such that *𝜎k > 𝜎* and

*𝑘* ≥ *𝑘̄* is a given iteration of [Algorithm 1](#_bookmark3). If there is an infinite sequence

*J*1 of successful iterations after *𝑘̄ ,* this leads to a contradiction with the

fact that *g* and *f* are bounded below.

In fact, since *𝜌* is a nondecreasing positive function, one has

*𝜌*(*𝜎k* ) ≥ *𝜌*(*𝜎*) *>* 0. Hence, if *𝑔*(*𝑥𝑘*+1 ) *< 𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* ) and *g*(*xk* ) *> C𝜌*(*𝜎k* ) for all *k* ∈ *J*1, then

*𝑔*(*𝑥𝑘*+1 ) *< 𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎*)*,*

there must exist an infinite subsequence *J*2*⊆J*1 of iterates for which which obviously contradicts the boundness below of *g* by 0. Thus

*𝑀* (*𝑥𝑘*+1 ) *< 𝑀* (*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* ). Hence,

*𝑀* (*𝑥𝑘*+1 ) *< 𝑀* (*𝑥𝑘* ) − *𝜌*(*𝜎*) for all *𝑘* ∈ *𝐽*2 *.*

Thus *M*(*xk* ) tends to -∞ which is a contradiction, since both *f* and *g* are

bounded below.

The proof is thus completed if there is an infinite number of success- ful iterations. However, if no more successful iterations occur after a

**Algorithm 2:** A globally convergent ES for general constraints (Restoration).

**Data**: Start from *𝑥𝑘𝑟* ∈ Ω*𝑢𝑟* given from the Main algorithm and

consider the same parameter as therein.

**1 for** *𝑘* = *𝑘𝑟 , 𝑘𝑟* + 1*, 𝑘𝑟* + 2*,* … **do**

**2 Step 1:** compute new sample points

sequences along certain limit directions (known as refining directions). By refining subsequence ([Audet and Dennis Jr., 2006](#_bookmark48)), we mean a subse- quence of unsuccessful iterates in the Main algorithm (see [Algorithm 1](#_bookmark3)) for which the step-size parameter converges to zero.

Assuming that *h* is Lipschitz continuous around the point *x*∗ ∈ Ω*ur*,

it is possible to use the Clarke-Jahn generalized derivative along a di-

rection *d*

*𝑌𝑘*+1 = {*𝑦*1

*𝑘*+1

*𝑘*+1

*,* … *, 𝑦𝜆*

}such that

*ℎ*◦(*𝑥* ; *𝑑*) = lim sup

*ℎ*(*𝑥* + *𝑡𝑑*) − *ℎ*(*𝑥*) *.*

*𝑖*

*𝑦*

*𝑘*+1

= *𝑥𝑘*

+ *𝜎𝑘 𝑑̃𝑖 , 𝑖* = 1*,* … *, 𝜆,*

∗ *𝑡*

*𝑥*→*𝑥*∗ *,𝑥*∈Ω*𝑢𝑟*

where the directions *𝑑̃𝑖* ’s are computed from the original ES directions *𝑑𝑖* ’s(which in turn are drawn from a chosen ES distribution C*𝑘* and scaled if necessary to

*𝑘*

*𝑘*

*𝑘*

satisfy*𝑑* ≤ *𝑑𝑖* ≤ *𝑑* );

*𝑡*↓0*,𝑥*+*𝑡𝑑*∈Ω*𝑢𝑟*

The latter derivative, proposed by [Jahn (1996)](#_bookmark38), can be seen as an adap- tation of the Clarke generalized directional derivative ([Clarke, 1983](#_bookmark57)) to

the presence of constraints. We note that definition of *h*∘(*x*∗ ; *d*) required

min

(*𝑦𝑖*

‖ *𝑘*‖

max

that *𝑥* + *𝑡𝑑* ∈ Ω for *x* ∈ Ω arbitrarily close to *x*∗ which can be guar-

*𝑢𝑟 𝑘*+1

**3**

), *𝑖* = 1*,* … *, 𝜆*, and reorder

theoffspring points in *𝑌𝑘*+1 = {*𝑦̃𝑘*+1 *,* … *, 𝑦̃𝑘*+1 }by increasing

**Step 2:** evaluate *𝑔*Ω

*𝑢𝑟*

*ur*

1 *𝜆*

order: *𝑔*Ω (*𝑦̃*1 ) ≤ ⋯ ≤ *𝑔*Ω (*𝑦̃𝜆* ).Select the new parents as

anteed if *d* is hypertangent to Ω*ur* at *x*∗ . In what comes next, *B*(*x*; *𝜖*) will

denote the closed ball formed by all points with a distance of no more

*𝑢𝑟*

*𝑘*+1

*𝑢𝑟*

*𝑘*+1 1 *𝜇*

than *𝜖* to *x*.

the best *𝜇* offspring sample points{*𝑦̃𝑘*+1 *,* … *, 𝑦̃𝑘*+1 }, and

compute their weighted mean

*𝜇*

= ∑ *𝜔 𝑦̃*

**Definition 3.1.** A vector *𝑑* ∈ ℝ*𝑛* is said to be a hypertangent vector to the set Ω*𝑢𝑟 ⊆* ℝ*𝑛* at the point *x* in Ω*ur* if there exists a scalar *𝜖 >* 0 such

*𝑡𝑟𝑖𝑎𝑙*

*𝑥*

*𝑘*+1

*𝑖 𝑖*

*𝑘 𝑘*+1

;

*𝑖*=1

that

*𝑦* + *𝑡𝑤* ∈ Ω*𝑢𝑟,* ∀*𝑦* ∈ Ω*𝑢𝑟* ∩ *𝐵*(*𝑥*; *𝜖*)*, 𝑤* ∈ *𝐵*(*𝑑*; *𝜖*)*,* and 0 *< 𝑡 < 𝜖.*

**Step 3: if** *𝑥𝑡𝑟𝑖𝑎𝑙* ∉ Ω*𝑢𝑟* **then**

*𝑘*+1

1. the iteration is declared unsuccessful;

# else

**6 if** *𝑔*(*𝑥𝑡𝑟𝑖𝑎𝑙* ) *< 𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* )⪯⪯⪯*and*⪯⪯⪯⪯*𝑔*(*𝑥𝑘* ) *> 𝐶𝜌*(*𝜎𝑘* )

*𝑘*+1

# then

1. the iteration is declared successful, set *𝑥𝑘*+1 = *𝑥𝑡𝑟𝑖𝑎𝑙* , and*𝜎𝑘*+1 ≥ *𝜎𝑘* (for example *𝜎𝑘*+1 = max{*𝜎𝑘 , 𝜎𝐸𝑆* });

*𝑘*

*𝑘*+1

# else

1. the iteration is declared unsuccessful;

# end

1. **end**
2. **if** *the iteration is declared unsuccessful* **then**

**13 if** *𝑀* (*𝑥𝑡𝑟𝑖𝑎𝑙* ) *< 𝑀* (*𝑥𝑘* ) **then**

*𝑘*+1

(starting at a new (*𝑘* + 1)−thiteration using *𝑥𝑘*+1 and **14** leave Restoration and return to the Main algorithm

The hypertangent cone to Ω*ur* at *x*, denoted by *𝑇 𝐻* (*𝑥*)*,* is the set of all

*𝑢𝑟*

Ω

hypertangent vectors to Ω*ur* at *x*. Then, the Clarke tangent cone to Ω*ur* at

*x* (denoted by *𝑇 𝐶𝐿*(*𝑥*)) can be defined as the closure of the hypertangent

Ω

*𝑢𝑟*

cone *𝑇 𝐻* (*𝑥*). The Clarke tangent cone generalizes the notion of tangent

Ω

*𝑢𝑟*

original definition *𝑑* ∈ *𝑇 𝐶𝐿*(*𝑥*) is given below. cone in Nonlinear Programming ([Nocedal and Wright, 2006](#_bookmark54)), and the

Ω

*𝑢𝑟*

**Definition 3.2.** A vector *𝑑* ∈ ℝ*𝑛* is said to be a Clarke tangent vector to the set Ω*𝑢𝑟 ⊆* ℝ*𝑛* at the point *x* in the closure of Ω*ur* if for every sequence

{*yk* } of elements of Ω*ur* that converges to *x* and for every sequence of

of vectors {*wk* } converging to *d* such that *𝑦𝑘* + *𝑡𝑘 𝑤𝑘* ∈ Ω*𝑢𝑟*. positive real numbers {*tk* } converging to zero, there exists a sequence

generalized derivative to Ω*ur* at *x*∗ as the limit For a direction *v* in the tangent cone, we consider the Clarke-Jahn

*ℎ*◦(*𝑥* ; *𝑣*) = lim *ℎ*◦(*𝑥* ; *𝑑*)

Ω*𝑢𝑟*

*𝜎𝑘*+1 );

# 15 else

∗ *𝑑*∈*𝑇 𝐻* (*𝑥*∗ )*,𝑑*→*𝑣* ∗

**16** set *𝑥𝑘*+1 = *𝑥𝑘* and*𝜎𝑘*+1 = *𝛽𝑘 𝜎𝑘* , with *𝛽𝑘* ∈ (*𝛽*1 *, 𝛽*2 );

# end

1. **end**
2. **Step 4:** update the ES step length *𝜎𝐸𝑆* , the distribution C*𝑘*+1 , and the weights(*𝜔*1 *,* … *, 𝜔𝜇* ) ∈ *𝑆*;

*𝑘*+1

*𝑘*+1

*𝑘*+1

(see [Audet and Dennis Jr., 2006](#_bookmark48)). A point *x*∗ ∈ Ω*ur* is considered Clarke

stationary if *h*∘(*x*∗ ; *d*) ≥ 0, ∀*𝑑* ∈ *𝑇 𝐶𝐿*(*𝑥*∗).

Ω*𝑢𝑟*

An important ingredient used in our convergence analysis is the no- tion of refining direction ([Audet and Dennis Jr., 2006](#_bookmark48)), associated with a convergent refining subsequence *K*. A refining direction is defined as

the limit point of {*ak* /  *ak * } for all *k* ∈ *K* suﬃciently large such that

# end

*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ∈ Ω*𝑢𝑟,* where *𝑎𝑘* =

*𝑖*=1

∑*𝜇*

*𝑘 𝑘*

*𝜔𝑖 𝑑̃𝑖* .

that one must have a subsequence of iterations driving *𝜎k* to zero. □ certain order, then this also leads to a contradiction. The conclusion is

**Theorem 3.1.** *Let f be bounded below and assuming that the restoration is not entered after a certain order.*

*There exists a subsequence K of unsuccessful iterates for which*

lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Moreover, if the sequence* {*xk* } *is bounded, there exists an*

The following convergence result concerns the determination of fea- sibility.

**Theorem 3.2.** *Let 𝑎𝑘* = *𝜇 𝜔𝑖 𝑑𝑖 and assume that f is bounded below. Suppose that the restoration is not entered after a certain order. Let x*∗ ∈ Ω*ur*

*𝑖*=1

*𝑘 𝑘*

∑

*for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Assume that g is Lipschitz continuous near x*∗ *with be the limit point of a convergent subsequence of unsuccessful iterates* {*xk* }*K constant 𝜈g >* 0*.*

*If 𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) *is a refining direction associated with* {*ak* /  *ak * }*K* , *then*

Ω*𝑢𝑟* ∘

*x*∗ *and a refining subsequence K*′ *such that* lim*𝑘*∈*𝐾 𝑥𝑘* = *𝑥*∗ *.*

unsuccessful iterates for which *𝜎𝑘*+1 goes to zero. In such a case we have **Proof.** From [Lemma 3.1](#_bookmark5), there must exist an infinite subsequence *K* of

*𝜎𝑘* = (1∕*𝛽𝑘* )*𝜎𝑘*+1 *, 𝛽k* ∈ (*𝛽*1 , *𝛽*2 ), and *𝛽*1 *>* 0, and thus *𝜎k* → 0, for *k* ∈ *K*,

*either 𝑔*(*𝑥*∗) = 0 *or g* (*x*∗ ; *d*) ≥ 0*.*

**Proof.** Let *d* be a limit point of {*ak* /  *ak * }*K* . Then, a subsequence *K*′ of *K* must exist such that *ak* /  *ak * → *d* on *K*′. On the other hand, we have for all *k*

too.

The second part of the theorem is proved by extracting a convergent

*𝑡𝑟𝑖𝑎𝑙*

*𝑘*+1

*𝑥*

*𝜇*

*𝑖 𝑖*

= ∑ *𝜔 𝑦̃*

*𝑘 𝑘*+1

*𝑖*=1

= *𝑥𝑘*

+ *𝜎𝑘*

*𝜇*

*𝑖 𝑖*

∑ *𝜔 𝑑*

*𝑘 𝑘*

*𝑖*=1

= *𝑥𝑘*

+ *𝜎𝑘 𝑎𝑘 ,*

subsequence *K*′ *⊂ K* for which *xk* converges to *x*∗ . □

Since the iteration *k* ∈ *K*′ is unsuccessful, *𝑔*(*𝑥𝑡𝑟𝑖𝑎𝑙*) ≥ *𝑔*(*𝑥* ) − *𝜌*(*𝜎* ) or

*𝑘*+1

*𝑘 𝑘*

Global convergence will be achieved by establishing that some type of directional derivatives are nonnegative at limit points of refining sub-

*g*(*xk* ) ≤ *C𝜌*(*𝜎k* ), and then either there exists an infinite number of the

first inequality or the second one as follows:

1. For the case where there exists a subsequence *K*1*⊆K*′ such that *g*(*xk* ) ≤ *C𝜌*(*𝜎k* ), it is trivial to obtain *𝑔*(*𝑥*∗) = 0 using both the con- tinuity of g and the fact that *𝜎k* tends to zero in *K*1.
2. For the case where there exists a subsequence *K*2*⊆K*′ such that the

**Proof.** See the proof of [(Gratton and Vicente, 2014, Theorem 4.2)](#_bookmark33). □

We point out that the assumption regarding the directions

{*ak* /  *ak * }*K* , in particular their density in the unit sphere, applies to a

sequence {*𝑎𝑘* ∕ *𝑎𝑘* }*𝐾*2

‖ ‖

Ω*𝑢𝑟*

quence {

*𝑎*

*𝜎* }

goes to zero in *K*

(*a*

is bounded above for

‖ ‖

converges to *𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) in *K*2 and the se-

given refining subsequence *K*″ and not to the whole sequence of iter-

ates. However, such a strengthening of the requirements on the density

*𝑘 𝑘 𝑘*∈*𝐾*2 2 *k*

all *k*, and so *𝜎k * *ak * tends to zero when *𝜎k* does). Thus one must have necessarily for *k* suﬃciently large in *K*2 , *𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ∈ Ω*𝑢𝑟* such

that

*𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) ≥ *𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* )*.*

From the definition of the Clarke-Jahn generalized derivative along

of the directions seems necessary for these types of directional meth- ods ([Audet and Dennis Jr., 2006](#_bookmark48)). By choosing the distribution C*𝑘* in the algorithm to be a multivariate normal distribution with mean zero

(the most commonly used choice in the literature), the density of the

particular for such choice of C*𝑘,* one has for any *𝑦* ∈ ℝ*𝑛* such that *𝑦* = 1 directions *ak* in the unit sphere is guaranteed with a probability 1. In and for any *𝛼* ∈ (0, 1), there exists a positive constant *𝜂* such that

‖ ‖

directions *𝑑* ∈ *𝑇 𝐻* (*𝑥*∗)*,*

Ω*𝑢𝑟*

*𝑔*◦(*𝑥*∗; *𝑑*) = lim sup

*𝑔*(*𝑥* + *𝑡𝑑*) − *𝑔*(*𝑥*)

*𝑡*

ℙ(cos(*𝐴𝑘* ∕ *𝐴𝑘 , 𝑦*) ≥ 1 − *𝛼, 𝐴𝑘* ≥ *𝜖*) ≥ *𝜂,*

where *Ak* is a random variable whose realization is *𝑎𝑘* = ∑*𝜇*

*𝑖*=1

‖ ‖ ‖ ‖

*𝜔𝑖 𝑑̃𝑖* .

*𝑥*→*𝑥*∗ *,𝑡*↓0*,𝑥*+*𝑡𝑑*∈Ω*𝑢𝑟*

*𝑘 𝑘*

≥ lim sup *𝑔*(*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*) − *𝑔*(*𝑥𝑘* )

*𝜎*

*𝑎*

[We now move to an intermediate optimality result. As in Gratton and](#_bookmark33)

The justification of such a claim is discussed in further detail in [Diouane et al. (2015a)](#_bookmark26).

*𝑘*∈*𝐾*2 *𝑘* ‖ *𝑘* ‖

[Vicente (2014), we will not use *x*∗ ∈ Ω explicitly in the proof but](#_bookmark33)

*𝑔*(*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖(*𝑎𝑘* ∕‖*𝑎𝑘* ‖)) − *𝑔*(*𝑥𝑘* ) ∘

[*qr*](#_bookmark33)

where,

= lim sup

*𝑘*∈*𝐾*2

*𝑎*

*𝜎*

*𝑘*‖ *𝑘*‖

– *𝑔𝑘 ,*

only *g* (*x*∗ ; *d*) ≤ 0. The latter inequality describes the cone of first order linearized directions under feasibility assumption *x*∗ ∈ Ω*qr*.

**Theorem 3.4.** *Let 𝑎𝑘* = ∑*𝜇 𝜔𝑖 𝑑𝑖 and assume that f is bounded below.*

*𝑖*=1

*𝑘 𝑘*

*𝑔* = *𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) − *𝑔*(*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*)

*Suppose that the restoration is not entered after a certain order.*

*Let x*∗ ∈ Ω*ur be the limit point of a convergent subsequence of unsuccessful*

*𝑘 𝜎*

*𝑎*

*𝑘*‖ *𝑘*‖

*iterates* {*xk* }*K for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Assume that g and f are Lipschitz*

*continuous near x*∗ *.*

from the Lipschitz continuity of *g* near *x*∗

*If 𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) *is a refining direction associated with* {*a* /  *a *}

*such*

*𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) − *𝑔*(*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*)

Ω*𝑢𝑟*

*that g* (*x*∗ ; *d*) ≤ 0*. Then f* (*x*∗ ; *d*) ≥ 0*.*

∘ ∘

*k k K*

*𝑔𝑘* =

*𝑎*

*𝜎*

*𝑘*‖ *𝑘*‖

**Proof.** By assumption there exists a subsequence *K*′*⊆K* such that the

*𝑎*

≤ *𝜈𝑔* ‖ − *𝑑*‖

*𝑘*

sequence {*𝑎𝑘* ∕‖*𝑎𝑘* ‖}*𝐾*′ converges to *𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) in *K*′ and the sequence

*𝑎𝑘*

‖ ‖ ‖

‖

{‖*𝑎* ‖*𝜎* } goes to zero in *K*

*𝑢𝑟*

*k*

Ω

*𝑘*

*𝑘 𝐾* ′

′, Thus one must have necessarily for

suﬃciently large in *K*′, *𝑥𝑡𝑟𝑖𝑎𝑙* = *𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ∈ Ω*𝑢𝑟*.

tends to zero on *K*2. Finally,

Since the iteration

*𝑘*+1

*𝑡𝑟𝑖𝑎𝑙*

*𝑔*◦(*𝑥* ; *𝑑*) ≥ lim sup *𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) − *𝑔*(*𝑥𝑘* ) + *𝜌*(*𝜎𝑘* ) − *𝜌*(*𝜎𝑘* )

*𝑘*

– *𝑔*

*𝑀* (*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* )*,* and thus

*k* ∈ *K*′ is unsuccessful, one has *𝑀* (*𝑥𝑘*+1 ) ≥

∗ *𝑘*∈*𝐾*2

*𝑎*

*𝑎*

(3)

*𝑎*

*𝜎*

*𝜎*

*𝜎*

*𝑘*‖ *𝑘*‖

*𝑘*‖ *𝑘*‖

*𝑓* (*𝑥* + *𝜎 𝑎* ) − *𝑓* (*𝑥* )

*𝑔*(*𝑥* + *𝜎 𝑎* ) − *𝑔*(*𝑥* )

*𝜌*(*𝜎* )

= lim sup

*𝑘*

*𝑘 𝑘*

*𝑘*

*𝑘*

*𝑘 𝑘*

*𝑘*

*𝑘*

*𝑘*∈*𝐾*2

*.*

*𝜎 𝑎*

‖ ‖

*𝑔*(*𝑥*

+ *𝜎 𝑎* ) − *𝑔*(*𝑥* ) + *𝜌*(*𝜎* )

*𝑘 𝑘 𝑘 𝑘 𝑘*

*𝑘*

*𝑘*

‖ *𝑘*‖ *𝑘*

‖ *𝑘*‖ *𝑘*

*𝑘*‖ *𝑘*‖

One then obtains *g*∘(*x*∗ ; *d*) ≥ 0. □

≥ −*𝛿̄*

–

*𝑎*

*𝜎*

*𝜎*

*𝑎*

Moreover, assuming that the set of the refining directions *𝑑* ∈

On the other hand,

*𝑓* ◦(*𝑥*∗; *𝑑*) = lim sup

*𝑥*→*𝑥*∗ *,𝑡*↓0*,𝑥*+*𝑡𝑑*∈Ω

*𝑓* (*𝑥* + *𝑡𝑑*) − *𝑓* (*𝑥*)

*𝑡*

*𝑇 𝐻* (*𝑥*∗)*,* associated with {*ak* /  *ak * }*K* , is dense in the unit sphere. One

Ω*𝑢𝑟*

≥ lim sup

*𝑓* (*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*) − *𝑓* (*𝑥𝑘* )

can show that the limit point *x*∗ is Clarke stationary for the flowing

′ *𝜎 𝑎*

*𝑘*∈*𝐾*  *𝑘* ‖ *𝑘* ‖

optimization problem, known as the constraint violation problem:

*𝑎*

*𝜎*

*𝑘*

= lim sup *𝑓* (*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖(*𝑎𝑘* ∕‖*𝑎𝑘* ‖)) − *𝑓* (*𝑥𝑘* ) − *𝑓 ,*

min *𝑔*(*𝑥*) (2)

*𝑠.𝑡. 𝑥* ∈ Ω*𝑢𝑟.*

where,

*𝑘*∈*𝐾* ′

*𝑘*‖ *𝑘*‖

**Theorem 3.3.** *Let 𝑎*

= ∑*𝜇*

*𝜔𝑖 𝑑𝑖 and assume that f is bounded below.*

*𝑓* = *𝑓* (*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) − *𝑓* (*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*) *,*

*Suppose that the restoration is not entered after a certain order. Assume that*

*𝑘*

*𝑖*=1

*𝑘 𝑘*

*the directions 𝑑̃𝑖 ’s and the weights 𝜔𝑖 ’s are such that (i) 𝜎k * *ak * *tends to zero*

*𝑘*

*𝜎*

*𝑎*

*𝑘 𝑘*

which then implies from [(3)](#_bookmark8)

*𝑘 𝑘*

‖ ‖

*when 𝜎k does, and (ii) 𝜌*(*𝜎k* )/(*𝜎k * *ak * ) *also tends to zero.*

*Let x*∗ ∈ Ω*ur be the limit point of a convergent subsequence of unsuccessful*

*𝑓* ◦(*𝑥* ; *𝑑*) ≥ lim sup *𝑓* (*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖(*𝑎𝑘* ∕‖*𝑎𝑘* ‖)) − *𝑓* (*𝑥𝑘* ) − *𝑓 ,*

*iterates* {*xk* }*K for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0 *and that 𝑇*Ω (*𝑥*∗) ≠ ∅*. Assume that g*

*𝐶𝐿*

′

*𝑘*∈*𝐾*  *𝑘* ‖ *𝑘* ‖

*is Lipschitz continuous near x*∗ *with constant 𝜈 >* 0

∗

*𝜎*

*𝑎*

*𝑘*

≥ lim sup −*𝛿̄ 𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) − *𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* )

– *𝑓*

*Then either (a) 𝑔*(*𝑥* ) = 0 *(implying x*∗ ∈ Ω

*and thus x*∗ ∈ Ω*) or (b)*

′ *𝑎 𝜎*

*𝜎 𝑎 𝑘*

∗ *qr*

*𝐶𝐿*

*𝑘*∈*𝐾*

‖ *𝑘*‖ *𝑘 𝑘*‖ *𝑘*‖

*if the set of refining directions 𝑑* ∈ *𝑇*Ω*𝑢𝑟* (*𝑥*∗) *associated with* {*𝑎𝑘* ∕‖*𝑎𝑘* ‖}*𝐾*′

*(where K*′ *is a subsequence of K for which 𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) ≥ *𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* )*)*

≥ lim sup −*𝛿*

*𝜎*

*𝑎*

+ *𝛿𝑔𝑘* − *𝜎*

*𝑎*

– *𝑓𝑘 ,*

*̄ 𝑔*(*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*) − *𝑔*(*𝑥𝑘* ) *̄*

*𝜌*(*𝜎𝑘* )

o

*is dense in 𝑇 𝐶𝐿*(*𝑥*∗) ∩ {*𝑑* ∈ ℝ*𝑛* ∶ ‖*𝑑*‖ = 1}*, then g* (*x*∗ ; *v*) ≥ 0 *for all 𝑣* ∈

Ω*𝑢𝑟*

Ω*𝑢𝑟*

where

*𝑇 𝐶𝐿*(*𝑥*∗) *and x*∗ *is a Clarke stationary point of the constraint violation prob-*

*𝑘*∈*𝐾* ′

*𝑘*‖ *𝑘*‖

*𝑘*‖ *𝑘*‖

*lem* [*(2)*](#_bookmark9)*.*

*𝑔* = *𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) − *𝑔*(*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*) *.*

*𝑘 𝜎*

*𝑎*

*𝑘*‖ *𝑘*‖

From the assumption *g*∘(*x*∗ ; *d*) ≤ 0, one has

*𝑔*(*𝑥𝑘* + *𝜎𝑘* ‖*𝑎𝑘* ‖*𝑑*) − *𝑔*(*𝑥𝑘* )

*𝑔*(*𝑥* + *𝑡𝑑*) − *𝑔*(*𝑥*)

1. Since at the unsuccessful iteration *k* ∈ *K*′, Restoration is never left, so one has *𝑀* (*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) ≥ *𝑀* (*𝑥𝑘* )*,* and the proof follows an argu-

lim sup

*𝑘*∈*𝐾* ′

*𝑎*

*𝜎*

*𝑘*‖ *𝑘*‖

≤ lim sup

*𝑥*→*𝑥*∗ *,𝑡*↓0*,𝑥*+*𝑡𝑑*∈Ω*𝑢𝑟 𝑡*

≤ 0*,*

ment already seen (see the proof of [Theorem 3.3](#_bookmark10)).

1. [The same proof as (Gratton and Vicente, 2014, Theo-](#_bookmark33)

one obtains then

*𝑓* ◦(*𝑥* ; *𝑑*) ≥ lim sup *𝛿̄𝑔*

*𝑎*

*𝑘*

– *𝜌*(*𝜎𝑘* )

– *𝑓 .* (4)

[rem 4.4). □](#_bookmark33)

*3.3. Case 2: the restoration algorithm is entered and left infinite times*

∗ *𝑘*∈*𝐾* ′

*𝑘 𝜎*

*𝑘*‖ *𝑘*‖

The Lipschitz continuity of both *g* and *f* near *x*∗ guaranties that the quan- tities *fk* and *gk* tend to zero in *K*′. Thus, the proof is completed since the right-hand-side of [(4)](#_bookmark11) tends to zero in *K*′. □

Finally, we derive the complete optimality result.

**Theorem 3.5.** *Assuming that f is bounded below and that Restoration is not*

**Theorem 3.7.** *Consider Algorithm* [*1*](#_bookmark3) *and assume that f is bounded below. Assume that Restoration is entered and left an infinite number of times.*

1. *Then there exists a refining subsequence.*
2. *Let x*∗ ∈ Ω*ur be the limit point of a convergent subsequence of unsuc- cessful of iterates* {*xk* }*K for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Assume that g is Lipschitz*

*continuous near x*∗ , *and let 𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) *be a corresponding refining direc-*

Ω*𝑢𝑟* ∘

*entered after a certain order.*

*Let x*∗ ∈ Ω*ur be the limit point of a convergent subsequence of unsuccessful iterates* {*xk* }*k* ∈ *K for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Assume that g and f are Lipschitz continuous near x*∗ *.*

*Assume that the set*

*tion. Then either 𝑔*(*𝑥*∗) = 0 *(implying x*∗ ∈ Ω*r and thus x*∗ ∈ Ω*) or g* (*x*∗ ;

*d*) ≥ 0*.*

1. *Let x*∗ ∈ Ω*ur be the limit point of a convergent subsequence of un- successful of iterates* {*xk* }*K for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Assume that g and f*

*are Lipschitz continuous near x*∗ , *and let 𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) *be a corresponding*

o Ω*𝑢𝑟*

*𝑇* (*𝑥*∗) = *𝑇 𝐻* (*𝑥*∗) ∩ {*𝑑* ∈ ℝ*𝑛* ∶ *𝑑* = 1*, 𝑔*◦(*𝑥*∗ *, 𝑑*) ≤ 0} (5)

Ω*𝑢𝑟*

‖ ‖

*has a non-empty interior.*

*Let the set of refining directions be dense in T*(*x*∗ )*. Then f*∘(*x*∗ , *v*) ≥ 0 *for all 𝑣* ∈ *𝑇 𝐶𝐿*(*𝑥*∗) *such that g*∘(*x*∗ , *v*) ≤ 0, *and x*∗ *is a Clarke stationary point*

Ω*𝑢𝑟*

*of the problem* [*(1)*](#_bookmark0)*.*

**Proof.** See the proof of [(Gratton and Vicente, 2014, Theorem 4.4)](#_bookmark33). □

Now, we provide the analysis of the two other cases, namely when

(a) an infinite run of consecutive steps inside Restoration or (b) one enters the restoration an infinite number of times.

* 1. *Case 2: the restoration algorithm is entered and never left*

In this case, by a refining subsequence below, we mean a subse- quence of unsuccessful Restoration iterates for which the step-size pa- rameter converges to zero.

**Theorem 3.6.** *Assume that f is bounded below and that the restoration is entered and never left.*

1. *Then there exists a refining subsequence.*
2. *Let x*∗ ∈ Ω*ur be the limit point of a convergent subsequence of unsuc- cessful of iterates* {*xk* }*K for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Assume that g is Lipschitz*

*continuous near x*∗ , *and let 𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) *be a corresponding refining direc-*

*refining direction such that g* (*x*∗ ; *d*) ≤ 0*. Then f*∘(*x*∗ ; *d*) ≥ 0*.*

(*iv*) *Assume that the interior of the set T*(*x*∗ ) *given in* [*(5)*](#_bookmark14) *is non-empty. Let the set of refining directions be dense in T*(*x*∗ )*. Then f*∘(*x*∗ , *v*) ≥ 0 *for all*

*𝑣* ∈ *𝑇 𝐶𝐿*(*𝑥*∗) *such that g*∘(*x*∗ , *v*) ≤ 0, *and x*∗ *is a Clarke stationary point.*

Ω*𝑢𝑟*

**Proof.** (*i*) Let *K*1*⊆K* and *K*2*⊆K* be two subsequences where Restoration

is entered and left respectively.

Since the iteration *k* ∈ *K*2 is unsuccessful in the Restoration, one knows that the step size *𝜎k* is reduced and never increased, one then obtains that *𝜎k* tends to zero. By assumption there exists a subsequence *K*′*⊆K*2 such that the sequence {*𝑎𝑘* ∕‖*𝑎𝑘* ‖}*𝑘*∈*𝐾*′ converges to *𝑑* ∈ *𝑇 𝐻* (*𝑥*∗)

Ω*𝑢𝑟*

1. For all *k* ∈ *K*′, one has *𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) ≥ *𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* ) or

in *K*2 and the sequence {‖*𝑎𝑘* ‖*𝜎𝑘* }*𝑘*∈*𝐾*′ goes to zero in *K*′.

*g*(*xk* ) ≤ *C𝜌*(*𝜎k* ), one concludes that either *𝑔*(*𝑥*∗) = 0 or *g*∘(*x*∗ ; *d*) ≥ 0.

1. For all *k* ∈ *K*′, one has *𝑀* (*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) ≥ *𝑀* (*𝑥𝑘* )*,* and from this we conclude that *f*∘(*x*∗ ; *d*) ≥ 0 if *g*∘(*x*∗ ; *d*) ≤ 0.
2. [The same proof as (Gratton and Vicente, 2014, Theo- rem 4.4). □](#_bookmark33)

To sum up, the analysis of the global convergence of [Algorithm 1](#_bookmark3) was provided depending on the number of times the restoration pro- cedure is entered. When the restoration is entered finite times, [Theorem 3.2](#_bookmark7) showed that the limit points of certain subsequences of it- erates are either feasible or Clarke stationary for the constraint violation problem [(2)](#_bookmark9). [Theorem 3.5](#_bookmark13) showed then that such limit points are Clarke stationary for the optimization problem [(1)](#_bookmark0). Our analysis provide sim-

o Ω*𝑢𝑟*

*tion. Then either 𝑔*(*𝑥*∗) = 0 *or g* (*x*∗ ; *d*) ≥ 0*.*

1. *Let x*∗ ∈ Ω*ur be the limit point of a convergent subsequence of un-* *successful of iterates* {*xk* }*K for which* lim*𝑘*∈*𝐾 𝜎𝑘* = 0*. Assume that g and f*

*are Lipschitz continuous near x*∗ , *and let 𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) *be a corresponding*

ilar feasibility and optimality results for the two remaining cases (i.e., when the restoration is “entered but never left” or “entered and left an infinite number of times”), see [Theorems 3.6](#_bookmark15) and [3.7](#_bookmark12).

*refining direction such that g*∘(*x*∗ ; *d*) ≤

Ω*𝑢𝑟*

0*. Then f*∘(*x*∗ ; *d*) ≥ 0*.*

# Numerical experiments

1. *Assume that the interior of the set T*(*x*∗ ) *given in* [*(5)*](#_bookmark14) *is non-empty. Let the set of refining directions be dense in T*(*x*∗ )*. Then f*∘(*x*∗ , *v*) ≥ 0 *for all*

*𝑣* ∈ *𝑇 𝐶𝐿*(*𝑥*∗) *such that g*∘(*x*∗ , *v*) ≤ 0, *and x*∗ *is a Clarke stationary point of*

Ω*𝑢𝑟*

*the problem* [*(1)*](#_bookmark0)*.*

**Proof.** (*i*) There must exist a refining subsequence *K* within this call

one has *𝑔*(*𝑥𝑘*+1 ) *< 𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* ) and *g*(*xk* ) *> C𝜌*(*𝜎k* ) for an infinite of the restoration, by applying the same argument of the case where

By assumption there exists a subsequence *K*′*⊆K* such that the se- subsequence of successful iterations (see the proof of [Theorem 3.1](#_bookmark5)). quence {*𝑎𝑘* ∕ *𝑎𝑘* }*𝑘*∈*𝐾*′ converges to *𝑑* ∈ *𝑇 𝐻* (*𝑥*∗) in *K*′ and the sequence

Ω*𝑢𝑟*

‖ ‖

{ *𝑎𝑘 𝜎𝑘* }*𝑘*∈*𝐾*′ goes to zero in *K*′. Thus one must have necessarily for *k*

‖ ‖

suﬃciently large in *K*′, *𝑥𝑡𝑟𝑖𝑎𝑙* = *𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ∈ Ω*𝑢𝑟*.

*𝑘*+1

(*ii*) Since the iteration *k* ∈ *K*′ is unsuccessful in the Restoration,

*𝑔*(*𝑥𝑘* + *𝜎𝑘 𝑎𝑘* ) ≥ *𝑔*(*𝑥𝑘* ) − *𝜌*(*𝜎𝑘* ) or *𝑔*(*𝑥𝑘*+1 ) ≤ *𝐶𝜌*(*𝜎𝑘* )*,* and the proof follows

an argument already seen (see the proof of [Theorem 3.2](#_bookmark7)).

In this section, we evaluate the performance of the proposed merit function approach using different solvers, different comparison proce- dures, and a large collection of non-linear constrained optimization problems. All the procedures were implemented in Matlab and run using Matlab 2019a on a MacBook Pro-2,4 GHz Intel Core i5, 4 GB RAM.

* 1. *Problems tested and testing strategies*

In what comes next, as a benchmark test, we will use 40 small- scale constrained test problems as given in [Audet et al. (2018)](#_bookmark43) (those problems are extracted from the CUTEst collection ([Gould et al., 2015](#_bookmark35)). The dimensions of the tested problems do not exceed 9 variables, with eventually bound constraints and no more than 13 nonlinear constraints (see [Audet et al., 2018](#_bookmark43), Table 1 for a detailed description on all the tested problems). For each test problem, the initial point provided by CUTEst

is used, the latter respects the bound contraints but does not necessarily satisfy the nonlinear constraints.

To illustrate the obtained results, we will use the two well-known testing strategies: data profiles [Moré and Wild (2009)](#_bookmark55) and performance profiles ([Dolan et al., 2006](#_bookmark30)). For data profiles, we use the following convergence test

Regarding the *𝛿̄* parameter, we tested 8 different values varied in range 10−2 and 105, see [Fig. 1](#_bookmark17)(b). The obtained profiles show that, for a small budget of evaluations, **ES-MF** is not sensitive to the value of *𝛿̄*. For

a larger budget, the performance changes slightly probably due to the

value of *𝛿̄* = 103 is shown to be very favorable to the **ES-MF** solver. stochastic nature of the solver. However, on the tested problems, one

0 0 Next, for the parameter *C*, we tested 8 different values varied in range

*𝑓*max − *𝑓*Ω(*𝑥*) ≥ (1 − *𝛼*)(*𝑓*max − *𝑓*min)*,*

while for the performance profiles, we make use of

*𝑓*Ω(*𝑥*) − *𝑓*min ≤ *𝛼*(*𝑓*min + 1)*,*

10−2 and 105, see [Fig. 1](#_bookmark17)(c). Again, the obtained profiles change slightly.

We suspect that the slight changes in the performance are just due to

the stochastic nature of the solver and consider that **ES-MF** is not very sensitive to the choice of the parameter *C*.

In what comes next, for the solver **ES-MF**, we set by default *𝛿̄* = 1*,*

where *𝛼* is the level accuracy and *𝑓* 0

max

represents the largest value

*𝐶* = 1*,* and use the 𝓁2-norm to define the constraint violation function

among all the feasible objective function values initially visited by all the *g*.

tested solvers (i.e., *𝑓* 0 = max*𝑠 𝑓* 0 where *𝑓* 0 represents the objective

max *𝑠 𝑠*

function value at the first feasible point visited by the solver *s*). The value

tolerance of 10−7 for constraint violation is used to consider a point as *f*min represents the best feasible solution found by the tested solvers. A

being feasible. We note that, if a solver fails to find a feasible starting point for a given problem, the problem is considered as unsolved, in this case the convergence test is not used. The performance and data profiles are computed for a maximum of 3000 function evaluations. For the stochastic solvers, we will describe our results using the median data/performance profile obtained over 20 runs.

* 1. *Implementation choices*

[Algorithm 1](#_bookmark3) and [Algorithm 2](#_bookmark6) are implemented in Matlab. The ob- tained implementation will be called **ES-MF**. Most of the parameter choices followed those in [Diouane et al. (2015b)](#_bookmark27) (where some of the user-specified parameters are the same used by directional direct search

methods and CMA-ES). In particular, the values of *𝜆* and *𝜇* and of

tion (see [Hansen, 2011](#_bookmark36)): *𝜆* = 4 + floor(3 log(*𝑛*))*, 𝜇* = floor(*𝜆*∕2)*,* where the initial weights are those of CMA-ES for unconstrained optimiza- floor( · ) rounds to the nearest integer, and *𝜔𝑖* = *𝑎𝑖* ∕(*𝑎*1 + ⋯ + *𝑎𝜇* )*, 𝑎𝑖* = log(*𝜆*∕2 + 1∕2) − log(*𝑖*)*, 𝑖* = 1*,* … *, 𝜇*. The choices of the distribution C*𝑘* and of the update of *𝜎𝐸𝑆* also followed CMA-ES for unconstrained

0

*𝑘*

tions, the forcing function selected was *𝜌*(*𝜎*) = 10−4 *𝜎*2. To reduce the optimization. As used in most directional direct search implementa- step length in unsuccessful iterations we used *𝜎𝑘*+1 = 0*.*9*𝜎𝑘* which cor- responds to setting *𝛽*1 = *𝛽*2 = 0*.*9. For successful iterations we set *𝜎𝑘*+1 = max{*𝜎𝑘 , 𝜎𝐶𝑀𝐴*−*𝐸𝑆* } (with *𝜎𝐶𝑀𝐴*−*𝐸𝑆* the CMA step size used in ES). The directions *𝑑𝑖 , 𝑖* = 1*,* … *, 𝜆,* were scaled if necessary to obey the safeguards

*𝑑𝑖*

≤

*𝑘*

*𝑘 𝑘*

* 1. *The extreme barrier versus the merit function for ES*

In this subsection, we present a comparison between **ES-MF** and **ES- EB** from [Diouane et al. (2015b)](#_bookmark27) (**ES-EB** can be seen as a particular in- stance of **ES-MF** where all the constraints are UR). Since the solver **ES-EB** requires a feasible starting point, when the starting point is infeasible, finding a feasible point is accomplished by minimizing the constraint violation function *g*.

ing two levels of accuracy 10−3 and 10−7 . One can see that the extreme [Fig. 2](#_bookmark18) depicts the resulting performance and data profiles consider-

barrier approach is not able to solve more than 50% of the problems (as shown by the performance profiles). The data profiles indicate that the extreme barrier can be competitive for small budgets. Overall, the merit function approach is outperforming the extreme barrier approach. Thus, relaxing the constraints clearly makes it possible to reach better optimal solutions which motivates the use of the merit function approach **ES-MF** instead of **ES-EB**.

* 1. *Comparison of solvers using the problems from the CUTEst collection*

To quantify the eﬃciency of **ES-MF**, we include in our numerical comparison the solvers **MADS-PB, DDS-MF**, and **CSA-AL**:

* + - **MADS-PB** [Audet and Dennis Jr. (2009)](#_bookmark49): a mesh adaptive direct search (MADS) method where a progressive barrier (PB) approach has been implemented ([Audet and Dennis Jr., 2009](#_bookmark49)) to handle QR constraints. The progressive barrier approach, proposed in MADS, enjoys similar convergence properties as for our algorithm, hence, a comparison between the two solvers is very meaningful. For the

*𝑑*min

‖ *𝑘*‖

≤ *𝑑*

max

*,* with *𝑑*

min

= 10−10 and *𝑑*

max

= 1010 . The initial step

MADS solver, we used the implementation given in the NOMAD

finite lower and upper bounds for a variable, then *𝜎*0 is set to the half size is estimated using only the bound constraints: If there is a pair of of the minimum of such distances, otherwise *𝜎*0 = 1.

* 1. *Sensitivity analysis*

The proposed evolution strategy introduces some user-specified con- trol parameters and their performances might depend on the setting of these parameters. A full sensitivity analysis of all the control parame- ters of the merit function approach can be computationally demanding and is beyond the scope of this paper. Hence, this subsection focuses on

parameters, namely, the constants *𝛿̄* and *C* as well as the choice of norm the sensitivity of **ES-MF** with respect to the newly introduced control

type used to evaluate *g*.

choices for the constants *𝛿̄* and *C* as well as for the norm type used to [Fig. 1](#_bookmark17) shows their performance and the data profiles using different

evaluate the constraint violation function *g*. With respect to the choice the norm in *g*, see [Fig. 1](#_bookmark17)(a), one can see that the use of 𝓁2-norm is clearly favorable to our approach in particular with a large budget of objective function evaluations. The choice of working with the 𝓁2-norm to evaluate *g* was shown to perform better for the progressive barrier approach used in MADS ([Audet and Dennis Jr., 2009](#_bookmark49)).

package ([Le Digabel, 2011](#_bookmark46)), version 3.9.1 (C++ version linked to Matlab via a mex interface). This solver is deterministic.

* **DDS-MF** [Gratton and Vicente (2014)](#_bookmark33): a Matlab implementation of a directional direct search (DDS) method where a merit function (MF) is used to handle QR constraints. The parameter choices followed those given in the numerical section of [Gratton and Vicente (2014)](#_bookmark33). We recall that **ES-MF** is inspired from the **DDS-MF** method, hence including the latter solver in the comparison can be also very mean- ingful. We note also that this is the first time **DDS-MF** is compared using an extensive test set. The behavior of the solver is stochastic

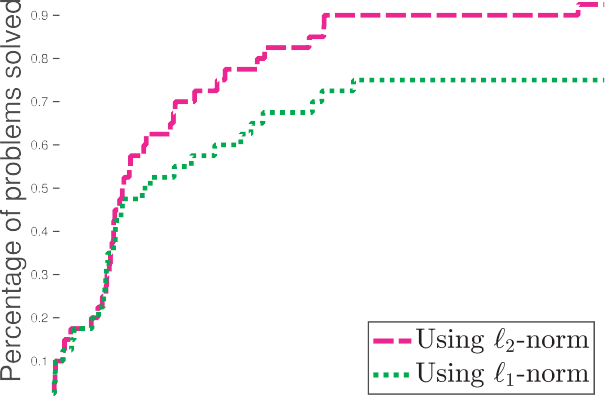
as it generates randomly (at most) *𝑛* + 1 directions at each iteration

of the algorithm.

* **CSA-AL** [Atamna et al. (2018)](#_bookmark40): a Matlab implementation of CMA-ES using an augmented Lagrangian approach to handle QR constraints. For the CMA-ES part, we used the same choice of parameters as for **ES-MF**, for the parameters associated with the augmented La- grangian part we chose the values given in [Atamna et al. (2018)](#_bookmark40).

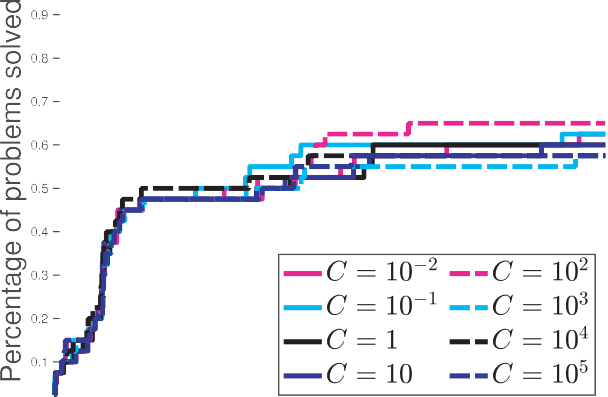
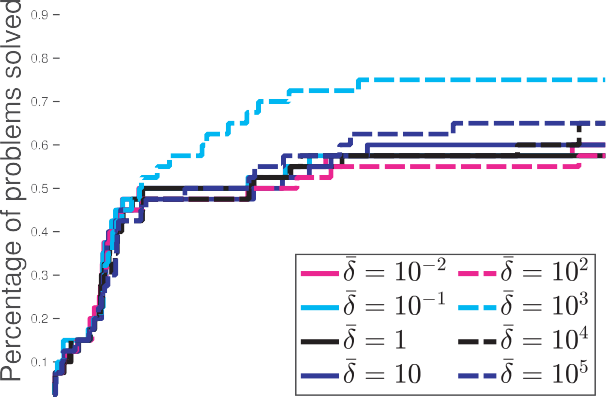
For all the solvers, we consider that all the nonlinear constraints are QR except the bounds which are treated using an 𝓁2-projection.

two accuracy levels 10−3 and 10−7 . Clearly, for all the runs, **CSA-AL** is [Fig. 3](#_bookmark19) reports the median (out of 20 runs) profiles considering the









**Fig. 1.** Median profiles for the solver **ES-MF** computed using 40 problems from the CUTEst set and different control parameters.









**Fig. 2.** Median profiles for the solvers **ES-MF** and **ES-EB** using 40 problems from the CUTEst set.

performing the worst among all the tested solvers. For the resulting data profiles, one can see that with a small budget, **DDS-MF** and **MADS-PB** exhibit better performance than the **ES-MF**. However, when the budget is getting larger, **ES-MF** performs the best. From the resulting perfor- mance profiles, one can see that in terms of eﬃciency (i.e., small values

of *𝜏*), **DDS-MF** is shown to be best. The **ES-MF** solver performs better in

terms of robustness (i.e., large values of *𝜏*).

In conclusion, first, clearly the **ES-MF** solver leads to very good re-

sults compared to **CSA-AL**. In fact, in our tests, **CSA-AL** showed diﬃ- culties finding feasible points while making progress on the objective function. We stress that the main difference between the two evolu- tion strategies is the restoration procedure, the latter helps **ES-MF** to progress better towards feasible zones without severe deterioration in terms of the objective function value. Second, **ES-MF** can be very com- petitive with both solvers **DDS-MF** and **MADS-PB**, in particular when using a large number of function evaluations.

* 1. *Comparison of solvers using global optimization test problems*

To confirm the results obtained when using CUTEst problems, we perform complementary tests using a set of problems with a diver-

sity of features and the kind of diﬃculties that appear in constrained [global optimization. The test set is that used in Hock and Schit-](#_bookmark39) [tkowski](#_bookmark52) [(1981),](#_bookmark39) [Koziel](#_bookmark44) [and Michalewicz (1999) and Michalewicz and](#_bookmark39)

[Schoenauer (1996) and comprises 12 well-known test problems (see](#_bookmark52) [Table 1](#_bookmark20)). The problems G2, G3, and G8 are originally maximization problems and were converted to minimization.

In addition to such problems, we include three realistic prob- lems. The first one is the tension-compression string (TCS) prob- lem ([Coello and Montes, 2002](#_bookmark25)), the aim is to minimize the weight of

a tension-compression string subject to constraints on minimum deflec- tion, shear stress, surge frequency, limits on outside diameter and on de- sign variables. The design variables are the mean coil diameter; the wire diameter and the number of active coils. The second problem is the well

known welded beam design (WBD) problem ([Coello and Montes, 2002](#_bookmark25))

where a welded beam is designed with a minimum cost subject to con- straints on shear stress; bending stress in the beam; buckling load on

the bar; end deflection of the beam; and side constraints. The third opti- mization problem is a multidisciplinary design optimization (MDO) prob- lem ([Gramacy and Digabel, 2015; Tribes et al., 2005](#_bookmark37)) where a simplified

wing design (built around a tube) is looked at. For this problem, one tries to minimize the range of the aircraft under coupled aero-structural









**Fig. 3.** Median profiles for the solvers **ES-MF, MADS-PB, DDS-MF**, and **CSA-AL**, using 40 problems from the CUTEst set.

**Table 1**

Description of the features of the 15 global optimization problems: the dimension *n*, the number of the QR constraints *m*, the number of the lower bounds # LB, the number of the upper bounds # UB, the initial objective value *f*(*x*0), the initial constraints violation *g*(*x*0), and the best known feasible solution *f*opt .

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Problem | *n* | *m* | # UB | # LB | *f*(*x*0 ) | *g*(*x*0 ) | *f*opt |
| G1 | 13 | 9 | 13 | 13 | −228*.*028 | 93357.8 | −15 |
| G2 | 20 | 2 | 20 | 20 | −0*.*0641952 | 0 | −0*.*803619 |
| G3 | 20 | 1 | 20 | 20 | −5*.*53267*𝑒* − 07 | 0.582395 | −1 |
| G4 | 5 | 6 | 5 | 5 | −24703*.*8 | 4.58618 | −30665*.*5 |
| G6 | 2 | 2 | 2 | 2 | 777287 | 1*.*78677*𝑒* + 08 | −6961*.*81 |
| G7 | 10 | 8 | 10 | 10 | 1154.69 | 410492 | 24.3062 |
| G8 | 2 | 2 | 2 | 2 | −6*.*40052*𝑒* − 09 | 4322.48 | −0*.*095825 |
| G9 | 7 | 4 | 7 | 7 | 156193 | 3*.*67173*𝑒* + 06 | 680.63 |
| G10 | 8 | 6 | 8 | 8 | 20711.3 | 6.01742 | 7049.33 |
| G11 | 2 | 1 | 2 | 2 | 4.97537 | 3.95049 | 0.75 |
| G12 | 3 | 1 | 3 | 3 | −0*.*532992 | 0 | −1 |
| G13 | 5 | 3 | 5 | 5 | 7.97186 | 71.9042 | 0.0539498 |
| TCS | 3 | 4 | 3 | 3 | 3*.*51385*𝑒* + 07 | 2*.*15037*𝑒* + 10 | 5868.76 |
| WBD | 4 | 6 | 4 | 4 | 278.59 | 1150.36 | 0.0126653 |
| MDO | 7 | 3 | 7 | 7 | −10*.*6934 | 2*.*3618*𝑒* + 07 | −16*.*61011 |



**Fig. 4.** Median profiles for the solvers **ES-MF, MADS-PB**, and **DDS-MF**, using 15 global optimization test problems.

constraints. The problem has 7 optimization variables corresponding to the geometry of the wing. The details of the three realistic problems features are included in [Table 1](#_bookmark20).

To allow the analysis of the asymptotic eﬃciency and the robustness of the tested solvers, we generate performance and data profile using a larger maximal number of function evaluation of 104. The starting point

*x*0 is chosen to be the same for all solvers and set to (*𝐿𝐵* + *𝑈𝐵*)∕2 where

*LB* are the lower bound constraints and *UB* are the upper bound con-

straints. We consider that all the constraints as QR except the bounds on the design variables which are treated using the 𝓁2-projection for all the

solvers. We note that problems G3, G11, and WBD contain equality con- straints. When a constraint is of the form *𝑐𝑒*(*𝑥*) = 0*,* we use the following relaxed inequality constraint instead *𝑐𝑖* (*𝑥*) = *𝑐𝑒*(*𝑥*) − 10−5 ≤ 0. We de-

*𝑖*

*𝑖*

| |

scribe our finding using the median performance and data profiles over 20 runs.

[Fig. 4](#_bookmark21) reports the obtained profiles for the solvers **MADS-PB, DDS- MF** and **ES-MF** using a maximal budget of 104. Additionally, we include the profiles of a variant of the solver **MADS-PB** where the variable neigh- borhood search (VNS) strategy is enabled to enhance its global perfor-

mance (by setting the flag vns\_search to 1 in the NOMAD package).

The latter solver is denoted by **MADS-PB (with VNS)** in [Fig. 4](#_bookmark21). We note

also that the solver **CSA-AL** is no longer included in the comparison as it displayed the worst results in our tests (it produced unfeasible solu- tions on most of the tested problems). Clearly, one can see that, unlike the previous test bed, the **ES-MF** solver outperforms the solvers **MADS- PB** and **DDS-MF**, particularly when considering a large function evalua-

tions. For the low accuracy level (i.e., *𝛼* = 10−3 ), enabling the VNS option

improves significantly the eﬃciency of **MADS-PB**. For such accuracy,

the solver **MADS-PB (with VNS)** reaches better eﬃciency performance compared to **ES-MF**. However, considering a higher accuracy level (i.e.,

*𝛼* = 10−7 ) tends to degrade the performance of **MADS-PB (with VNS)**

compared to **ES-MF**.

[Tables 2](#_bookmark22) and [3](#_bookmark23) depict the final obtained results for the solvers **MADS- PB, DDS-MF, MADS-PB (with VNS)** and **ES-MF**, using a maximal bud- get of 104 function evaluations. For each problem, we display the opti-

mal objective value found by the solver *f*(*x*∗ ), the associated constrained violation *g*(*x*∗ ), and the number of objective function evaluations #*𝑓*

needed to reach *x*∗ . When a solver returns a flag error or encounters an internal problem, we display “∗”. At the solution *x*∗ , one requires at least a tolerance of 10−5 on the constraint violation to consider *x*∗

as feasible with respect to QR constraints. Considering the median run,

**ES-MF** converged to a feasible solution for all the problems, **MADS-PB**

**Table 2**

Obtained results with **MADS-PB** and **DDS-MF**, using 15 global optimization test problems.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pb | *f*(*x*∗ ) |  |  |  | #*𝑓* |  |  |  | *g*(*x*∗ ) |  | | |
|  | Best | Median | Worst |  | Best | Median | Worst |  | Best | Median | Worst |  |
| **MADS-PB** | | | | | | | | | | | | |
| G1 | −12.4531 | −12.4531 | −12.4531 | 4202 | | 4202 | 4202 | 2e−26 | | 2e−26 | 2e−26 | |
| G2 | −0.321533 | −0.321533 | −0.321533 | 8194 | | 8194 | 8194 | 0 | | 0 | 0 | |
| G3 | −0.00101297 | −0.00101297 | −0.00101297 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G4 | −30665.5 | −30665.5 | −30665.5 | 1846 | | 1846 | 1846 | 8.5e−27 | | 8.5e−27 | 8.5e−27 | |
| G6 | −6961.81 | −6961.81 | −6961.81 | 427 | | 427 | 427 | 7.3e−27 | | 7.3e−27 | 7.3e−27 | |
| G7 | 30.0027 | 30.0027 | 30.0027 | 2161 | | 2161 | 2161 | 2.9e−26 | | 2.9e−26 | 2.9e−26 | |
| G8 | −0.095825 | −0.095825 | −0.095825 | 350 | | 350 | 350 | 0 | | 0 | 0 | |
| G9 | 680.915 | 680.915 | 680.915 | 1769 | | 1769 | 1769 | 5e−27 | | 5e−27 | 5e−27 | |
| G10 | 7973.6 | 7973.6 | 7973.6 | 10,000 | | 10,000 | 10,000 | 4.5e−06 | | 4.5e−06 | 4.5e−06 | |
| G11 | 0.7499 | 0.7499 | 0.7499 | 9355 | | 9355 | 9355 | 1e−26 | | 1e−26 | 1e−26 | |
| G12 | −1 | −1 | −1 | 425 | | 425 | 425 | 0 | | 0 | 0 | |
| G13 | 0.679994 | 0.679994 | 0.679994 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| TCS | ∗ | ∗ | ∗ | ∗ | | ∗ | ∗ | ∗ | | ∗ | ∗ | |
| WBD | 2.21815 | 2.21815 | 2.21815 | 3625 | | 3625 | 3625 | 1e−26 | | 1e−26 | 1e−26 | |
| MDO | −16.6007 | −16.6007 | −16.6007 | 6837 | | 6837 | 6837 | 0 | | 0 | 0 | |
| **DDS-MF** |  |  |  |  |  |  |  |  |  |  |  |  |
| G1 | −14.6929 | −11.8944 | −7.76563 | 4529 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G2 | −0.268315 | −0.195197 | −0.174585 | 8237 | | 9364 | 10,000 | 0 | | 0 | 0 | |
| G3 | −0.245346 | −0.000195272 | −0 | 980 | | 10,000 | 10,000 | 0 | | 0 | 2.8e−05 | |
| G4 | −32217.4 | −29246.5 | −23837.1 | 10,000 | | 10,000 | 10,000 | 0 | | 0.7 | 6 | |
| G6 | −7495.49 | −7331.06 | −7206.23 | 10,000 | | 10,000 | 10,000 | 0.023 | | 0.054 | 0.11 | |
| G7 | 24.8165 | 26.2708 | 30.9808 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G8 | −0.095825 | −0.095825 | −0.0258078 | 285 | | 324 | 10,000 | 0 | | 0 | 0 | |
| G9 | 681.499 | 683.972 | 691.198 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 9.3e−07 | |
| G10 | 3714.74 | 6463.86 | 8790.21 | 6079 | | 10,000 | 10,000 | 0.014 | | 0.086 | 0.44 | |
| G11 | 0.748826 | 0.749978 | 0.750995 | 10,000 | | 10,000 | 10,000 | 0 | | 4.7e−08 | 1.2e−06 | |
| G12 | −0.986446 | −0.554001 | −0.553667 | 10,000 | | 10,000 | 10,000 | 0 | | 2.2e−10 | 5.8e−08 | |
| G13 | 0.0932763 | 0.903758 | 8.50155 | 10,000 | | 10,000 | 10,000 | 0 | | 3.7e−08 | 1 | |
| TCS | 0.0154595 | 0.0514077 | 0.0547682 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 2.8e−06 | |
| WBD | 2.26572 | 4.03345 | 24.2009 | 684 | | 2103 | 10,000 | 0 | | 0 | 39 | |
| MDO | −15.8881 | −15.3359 | −14.0585 | 585 | | 1028 | 1738 | 0 | | 0 | 0 | |

**Table 3**

Obtained results with **ES-MF** and **MADS-PB (with VNS)**, using 15 global optimization test problems.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pb | *f*(*x*∗ ) |  |  |  | #*𝑓* |  |  |  | *g*(*x*∗ ) |  | | |
| ES-MF | Best | Median | Worst |  | Best | Median | Worst |  | Best | Median | Worst |  |
| G1 | −15.0003 | −15.0003 | −12.4537 |  | 10,000 | 10,000 | 10,000 |  | 1.2e−07 | 1.7e−07 | 5.6e−07 |  |
| G2 | −0.756445 | −0.716013 | −0.252014 |  | 5851 | 10,000 | 10,000 |  | 0 | 0 | 1.9e−10 |  |
| G3 | −1.00565 | −1.00538 | −1.03027 |  | 10,000 | 10,000 | 10,000 |  | 0 | 2.7e−06 | 3e−06 |  |
| G4 | −30665.5 | −30664.8 | −30649.1 |  | 10,000 | 10,000 | 10,000 |  | 0 | 0 | 9.6e−05 |  |
| G6 | −7865.39 | −6953.54 | −6369.01 |  | 4493 | 8406 | 10,000 |  | 0 | 1.4e−06 | 9.7e−05 |  |
| G7 | 24.3035 | 24.3037 | 24.3062 |  | 10,000 | 10,000 | 10,000 |  | 1.1e−08 | 1.3e−08 | 1.5e−06 |  |
| G8 | −0.095825 | −0.095825 | −0.0273164 |  | 1492 | 1653 | 10,000 |  | 0 | 0 | 2.7e−08 |  |
| G9 | 680.629 | 680.629 | 680.629 |  | 7231 | 8526 | 10,000 |  | 3.6e−07 | 3.6e−07 | 3.6e−07 |  |
| G10 | 7086.26 | 11177.6 | 18860.8 |  | 7288 | 9899 | 10,000 |  | 0 | 4.3e−05 | 9.4e-05 |  |
| G11 | 0.7499 | 0.7499 | 0.7499 |  | 2830 | 3522 | 10,000 |  | 1.6e-09 | 2.5e-09 | 3.9e-07 |  |
| G12 | −1 | −0.960558 | −0.783887 |  | 1457 | 3533 | 4281 |  | 0 | 1.6e−09 | 8.6e−09 |  |
| G13 | 0.0539573 | 0.438745 | 1 |  | 5465 | 10,000 | 10,000 |  | 1.2e−16 | 1.8e−09 | 3.2e−08 |  |
| TCS | 0.0126649 | 0.0126688 | 0.0132221 |  | 6598 | 10,000 | 10,000 |  | 1.1e−12 | 1.8e−10 | 6.9e−10 |  |
| WBD | 2.19747 | 2.21258 | 2.53771 |  | 8488 | 10,000 | 10,000 |  | 2.6e−10 | 2.8e−08 | 1.7e−08 |  |
| MDO | −16.612 | −16.612 | −16.6119 |  | 5031 | 10,000 | 10,000 |  | 0 | 0 | 1.1e−14 |  |
| **MADS-PB (with VNS)** | | | | | | | | | | | | |
| G1 | −15 | −15 | −15 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G2 | −0.697381 | −0.697381 | −0.697381 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G3 | −0.0870995 | −0.0870995 | −0.0870995 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G4 | −30665.5 | −30665.5 | −30665.5 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G6 | −6961.81 | −6961.81 | −6961.81 | 6523 | | 6523 | 6523 | 3.2e−27 | | 3.2e−27 | 3.2e−27 | |
| G7 | 24.8226 | 24.8226 | 24.8226 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G8 | −0.095825 | −0.095825 | −0.095825 | 6505 | | 6505 | 6505 | 0 | | 0 | 0 | |
| G9 | 680.632 | 680.632 | 680.632 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G10 | 7087.99 | 7087.99 | 7087.99 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G11 | 0.7499 | 0.7499 | 0.7499 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G12 | −1 | −1 | −1 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| G13 | 0.781443 | 0.781443 | 0.781443 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |
| TCS | ∗ | ∗ | ∗ | ∗ | | ∗ | ∗ | ∗ | | ∗ | ∗ | |
| WBD | 2.21815 | 2.21815 | 2.21815 | 10,000 | | 10,000 | 10,000 | 1e−26 | | 1e−26 | 1e−26 | |
| MDO | −16.6054 | −16.6054 | −16.6054 | 10,000 | | 10,000 | 10,000 | 0 | | 0 | 0 | |

converged as well to a feasible point for all the problems, except the TCS problem for which **MADS-PB** returns a flag error. The **DDS-MF** solver could not converge to a feasible solution for three problems G2, G4, and G5. In terms of the objective function value, one can see clearly

that **ES-MF** is outperforming both solvers **MADS-PB** and **DDS-MF**. As ex- pected, in terms of function evaluations, **MADS-PB** required in general less function evaluations than **ES-MF** to converge to a solution (but not necessarily better then the one found by **ES-MF**). The use of the vari- able neighborhood search option within MADS improves significantly its performance, **MADS-PB (with VNS)** is displaying similar performances compared to the **ES-MF**.

# Conclusion

In this paper, we proposed a globally convergent class of ES algo- rithms where a merit function is used to decide and control the dis- tribution of the generated points. The proposed approach included a restoration procedure which is entered whenever a decrease on the con- straint violation function is achieved while the objective function is being considerably increased. The obtained algorithm generalized the work ([Diouane et al., 2015b](#_bookmark27)) by including quantifiable relaxable con- straints. In the spirit of what is achieved in [Gratton and Vicente (2014)](#_bookmark33), the proposed convergence analysis was organized depending on the number of times Restoration is entered.

We provided numerical tests on problems from the CUTEst collection and a global optimization test bed. The results showed the potential of the proposed merit approach compared to existing direct search DFO solvers, in particular when using a large number of function evaluations.

# References

[Atamna, A., Auger, A., Hansen, N., 2018. On invariance and linear convergence of evolu- tion strategies with augmented Lagrangian constraint handling. Theor. Comput. Sci. 0, 1–53.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0001)

[Audet, C., Conn, A.R., Digabel, S.L., Peyrega, M., 2018. A progressive barrier deriva- tive-free trust-region algorithm for constrained optimization. Comput. Optim. Appl. 71 (2), 307–329.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0002)

[Audet, C., Hare, W., 2017. Derivative-Free and Blackbox Optimization. Springer Interna- tional Publishing, Cham, Switzerland.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0003)

[Audet, C., Dennis Jr., J.E., 2006. Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. 17, 188–217.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0004)

[Audet, C., Dennis Jr., J.E., 2009. A progressive barrier for derivative-free nonlinear pro- gramming.. SIAM J. Optim. 20 (1), 445–472.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0005)

[Auger, A., Brockhoff, D., Hansen, N., 2013. Benchmarking the local metamodel CMA-ES on the noiseless BBOB’2013 test bed. In: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation. ACM, New York, NY, USA,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0006)

[pp. 1225–1232.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0006)

[Auger, A., Hansen, N., Perez, Z.J., Ros, R., Schoenauer, M., 2009. Experimental compar- isons of derivative free optimization algorithms. In: 8th International Symposium on Experimental Algorithms. Springer Verlag, pp. 3–15.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0007)

[Bouzarkouna, Z., 2012. Well Placement Optimization. University Paris-Sud - Laboratoire de Recherche en Informatique.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0008)

[Clarke, F.H., 1983. Optimization and Nonsmooth Analysis. John Wiley & Sons, New York](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0009).

[Reissued by SIAM, Philadelphia, 1990](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0009)

Coello, C. A. C.,. List of references on constraint-handling techniques used with evolution- ary algorithms. <https://www.cs.cinvestav.mx/~constraint/constbib.pdf>.

[Coello,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0010) [C.A.C., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput. Methods Appl. Mech. Eng. 91, 1245–1287.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0010)

[Coello,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0011) [C.A.C., Montes, E.M., 2002. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv. Eng. Inf. 16, 193–203.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0011)

[Conn,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0012) [A.R., Scheinberg, K., Vicente, L.N., 2009. Introduction to Derivative-Free Optimiza- tion. SIAM, Philadelphia.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0012)

[Diouane,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0013) [Y., 2014. Globally Convergent Evolution Strategies With Application to Earth Imaging Problem in Geophysics. Institut National Polytechnique de Toulouse.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0013)

[Diouane,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0014) [Y., Gratton, S., Vasseur, X., Vicente, L.N., Calandra, H., 2016. A parallel evolution strategy for an earth imaging problem in geophysics. Optim. Eng. 17, 3–26.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0014)

[Diouane,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0015) [Y., Gratton, S., Vicente, L.N., 2015. Globally convergent evolution strategies.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0015)

[Math. Program. 152, 467–490.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0015)

[Diouane,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0016) [Y., Gratton, S., Vicente, L.N., 2015. Globally convergent evolution strategies for constrained optimization. Comput. Optim. Appl. 62, 323–346.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0016)

[Dolan,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0017) [E.D., Moré, J.J., Munson, T.S., 2006. Optimality measures for performance profiles.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0017)

[SIAM J. Optim. 16, 891–909.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0017)

[Fletcher,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0018) [R., Leyffer, S., 2002. Nonlinear programming without a penalty function. Math.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0018)

[Program. 91, 239–269.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0018)

[Gould,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0019) [N.I.M., Orban, D., Toint, P.L., 2015. CUTEst: a constrained and unconstrained test- ing environment with safe threads for mathematical optimization.. Comput. Optim. Appl. 60, 545–557.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0019)

[Gramacy,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0020) [R.B., Digabel, S.L., 2015. The mesh adaptive direct search algorithm with treed gaussian process surrogates. Pac. J. Optim. 11, 719–747.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0020)

[Gratton,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0021) [S., Vicente, L.N., 2014. A merit function approach for direct search. SIAM J. Optim. 24, 1980–1998.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0021)

Hansen, [N., 2011. The CMA Evolution Strategy: A tutorial. Available at https://www. lri.fr/~hansen/cmatutorial.pdf](https://www.lri.fr/~hansen/cmatutorial.pdf)

[Hansen,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0023) [N., Auger, A., Raymond, R.R., Finck, S., Pošík, P., 2010. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: Proceed- ings of the 12th Annual Conference Companion on Genetic and Evolutionary Compu- tation. ACM, New York, NY, USA, pp. 1689–1696.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0023)

[Hansen,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0024) [N., Ostermeier, A., Gawelczyk, A., 1995. On the adaptation of arbitrary normal mutation distributions in evolution strategies: the generating set adaptation. In: Es- helman, L. (Ed.), Proceedings of the Sixth International Conference on Genetic Algo- rithms, Pittsburgh, pp. 57–64.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0024)

[Hock,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0025) [W., Schittkowski, K., 1981. Test Examples for Nonlinear Programming Codes.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0025)

[Springer-Verlag New York, Inc., Secaucus, NJ, USA.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0025)

[Jahn,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0026) [J., 1996. Introduction to the Theory of Nonlinear Optimization. Springer-Verlag, Berlin.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0026)

[Kolda,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0027) [T.G., Lewis, R.M., Torczon, V., 2003. Optimization by direct search: new perspec- tives on some classical and modern methods. SIAM Rev. 45, 385–482.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0027)

[Koziel,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0028) [S., Michalewicz, Z., 1999. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol. Comput. 7, 19–44.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0028)

[Kramer,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0029) [O., 2010. A review of constraint-handling techniques for evolution strategies.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0029)

[Appl. Comp. Intell. Soft Comput. 2010, 1–19.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0029)

[Le](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0030) [Digabel, S., 2011. Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. 37, 1–15.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0030)

[Le](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0031) [Digabel, S., Wild, S., 2015. A Taxonomy of Constraints in Simulation-Based Optimiza- tion. Technical Report. Les cahiers du GERAD.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0031)

[Martínez,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0032) [J.M., Sobral, F.N.C., 2013. Constrained derivative-free optimization on thin do- mains. J. Glob. Optim. 56 (3), 1217–1232.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0032)

[Michalewicz,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0033) [Z., Schoenauer, M., 1996. Evolutionary algorithms for constrained parame- ter optimization problems. Evol. Comput. 4, 1–32.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0033)

[Moré,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0034) [J.J., Wild, S.M., 2009. Benchmarking derivative-free optimization algorithms. SIAM](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0034)

[J. Optim. 20, 172–191.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0034)

[Nocedal,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0035) [J., Wright, S.J., 2006. Numerical Optimization, second ed. Springer, New York, NY.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0035)

[Rechenberg,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0036) [I., 1973. Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzip- ien der Biologischen Evolution. Frommann-Holzboog.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0036)

[Rios,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0037) [L., Sahinidis, N., 2010. Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Glob. Optim. 56, 1247–1293.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0037)

[Tribes,](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0038) [C., Dubé, J.-F., Trépanier., J.-Y., 2005. Decomposition of multidisciplinary opti- mization problems: formulations and application to a simplified wing design. Optim. Eng. 37, 775–796.](http://refhub.elsevier.com/S2192-4406(20)30001-0/sbref0038)