EURO Journal on Computational Optimization 9 (2021) 100008

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

journal homepage: www.elsevier.com/locate/ejco

Pareto front approximation through a multi-objective augmented Lagrangian method

Guido Cocchi[a], Matteo Lapucci[b,*], Pierluigi Mansueto[b]

[a] Intuendi Srl, Via Vittorio Emanuele II 165, Firenze 50139, Italy
[b] DINFO, Università di Firenze, Via di Santa Marta 3, Firenze 50139, Italy

ARTICLE INFO

2020 MSC:
90C29
90C30

Keywords:
Multi-objective optimization
Augmented Lagrangian method
Pareto front approximation
Pareto stationarity
Global convergence

ABSTRACT

In this manuscript, we consider smooth multi-objective optimization problems with convex constraints. We propose an extension of a multi-objective augmented Lagrangian Method from recent literature. The new algorithm is specifically designed to handle sets of points and produce good approximations of the whole Pareto front, as opposed to the original one which converges to a single solution. We prove properties of global convergence to Pareto stationarity for the sequences of points generated by our procedure. We then compare the performance of the proposed method with those of the main state-of-the-art algorithms available for the considered class of problems. The results of our experiments show the effectiveness and general superiority w.r.t. competitors of our proposed approach.

1. Introduction

Multi-objective optimization is a mathematical tool which proved to be particularly suited to model and tackle real-world problems where many contrasting goals have to be reached. Successful applications of multi-objective optimization can be found, for example, in statistics (Carrizosa and Frenk, 1998), design (Fu and Diwekar, 2004; Jüschke et al., 1997; Shan and Wang, 2005), engineering (Campana et al., 2018; Kasperska et al., 2004; Liuzzi et al., 2003; Pellegrini et al., 2014; Sun et al., 2016), environmental analysis (Fliege, 2001; Leschine et al., 1992), management science (Gravel et al., 1992; White, 1998) or space exploration (Palermo et al., 2003; Tavana, 2004).

Popular classes of algorithms to solve multi-objective problems are those of scalarization methods (Drummond et al., 2008; Eichfelder, 2009; Fliege, 2004; Gass and Saaty, 1955; Geoffrion, 1968; Pascoletti and Serafini, 1984; Steuer and Choo, 1983; Zadeh, 1963) and of heuristic methods based on genetic and evolutionary strategies (Deb et al., 2002; Konak et al., 2006; Laumanns et al., 2002; Mostaghim et al., 2007). However, both these families of approaches come with shortcomings. Indeed, scalarization techniques require a detailed analysis of the problem structure in order to identify the weights defining a suitable scalarized objective. Moreover, an unfortunate choice of the weights may lead to unbounded scalar problems, even under strong regularity assumptions (Fliege et al., 2009, sec. 7). On the other hand, convergence properties cannot be stated for heuristic algorithms.

In order to overcome these limitations, descent methods extending classical scalar optimization techniques have been proposed to address constrained and unconstrained multi-objective problems (see, e.g., Drummond and Iusem, 2004; Fliege et al., 2009; Fliege and Svaiter, 2000). In this work we will bring particular attention to one of such algorithms, the extension of scalar augmented Lagrangian method (Birgin and Martinez, 2014) to the multi-objective case proposed by Cocchi and Lapucci (2020).

This group of algorithms typically produces, similarly to the scalar case, a sequence of points that is asymptotically driven to optimality. However, in the context of multi-objective applications, it is in practice crucial to generate a set of solutions constituting an approximation of the Pareto set, so that the user can choose, a posteriori, the solution providing the most appropriate trade-off among many.

Some recent works actually focused on strategies allowing to handle sequences of sets of points, instead of sequences of points, within multi-objective descent methods. This idea was first explored for derivative-free methods (Custódio et al., 2011; Liuzzi et al., 2016) and then considered for derivative based methods, both in the constrained (Fliege and Vaz, 2016) and the unconstrained (Cocchi et al., 2020) case.

The contribution of this paper consists of the definition of an extended version of the augmented Lagrangian algorithm for multi-objective optimization (ALAMO) proposed by Cocchi and Lapucci (2020), which deals with sets of points and effectively produces an approximation of the Pareto front for constrained vector-valued problems. The key elements that characterize the proposed algorithm are

* Corresponding author.
E-mail address: matteo.lapucci@unifi.it (M. Lapucci).

https://doi.org/10.1016/j.ejco.2021.100008
Received 15 February 2021; Received in revised form 29 June 2021; Accepted 13 August 2021

i) the management of a set of points at each iteration, as proposed by Custódio et al. (2011), which are all mutually nondominated w.r.t. the current augmented Lagrangian;
ii) the use of an Armijo-type line search, which possibly considers descent w.r.t. only a subset of objectives, in order to enrich the approximate front;
iii) the use of a common penalty parameter and Lagrange multipliers for all points in the set of solutions;
iv) the use of the multi-objective steepest descent algorithm from Fliege and Svaiter (2000) to make each point in the current set approximately Pareto-stationary w.r.t. the augmented Lagrangian, with increasing accuracy throughout the iterations.

For the proposed algorithm, we prove properties of convergence to Pareto-stationarity for the generated sequence of sets of points, without the need to recur to the concept of linked sequence introduced by Liuzzi et al. (2016). In fact, the convergence along linked sequences is implied by our result.

To the best of our knowledge, the SQP procedure by Fliege and Vaz (2016) is the only other derivative based method for constructing an approximated Pareto front of constrained multi-objective problems that can be found in the literature. It is worth remarking that, in contrast with the SQP method, convergence of our algorithm does not depend on a final refinement step that follows a finite exploration phase. As also noted by its authors, SQP can indeed be seen as a single point procedure run in a multi-start fashion. On the contrary, in our procedure convergence and exploration advance alongside, with both asymptotically improving.

The rest of the manuscript is organized as follows: in Section 2, we introduce basic concepts and notation that will be used in the presentation of the proposed procedure; in Section 3 we describe in detail our approach; we then provide the convergence analysis in Section 4. In Section 5, we show the result of computational experiments highlighting the good performance of the proposed procedure compared to a set of different state-of-the-art approaches. We finally give some concluding remarks in Section 6.

2. Preliminaries

In this paper, we consider optimization problems of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & F(x) = \big(f_1(x), \dots, f_m(x)\big)^T \\ \text{s.t.} \quad & g(x) \leq 0, \end{aligned} \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable function and $g : \mathbb{R}^n \to \mathbb{R}^p$ is a continuously differentiable, component-wise convex function, so that the feasible set $\Omega = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$ is a closed convex set, which we assume to be nonempty. We denote by J_F and J_g the Jacobian matrices associated respectively with F and g. In the following, we will also denote by e the vector of all ones. Note that equality constraints can be equivalently expressed as couples of opposite inequality constraints, so this formulation is in fact general. Actually, specific management of equality constraints can often be convenient from a computational perspective; the following discussion could easily be extended to address the presence of explicit equality constraints, but we prefer not to take them into account for the sake of simplicity.

In the following we will make use of a partial ordering of points in $\mathbb{R}^m$. Given two vectors $u, v \in \mathbb{R}^m$, we have

$$\begin{aligned} u < v \quad &\Leftrightarrow \quad u_i < v_i \quad \forall\, i = 1, \dots, m, \\ u \leq v \quad &\Leftrightarrow \quad u_i \leq v_i \quad \forall\, i = 1, \dots, m. \end{aligned}$$

We also say that *u dominates* v, and denote it by $u \lneqq v$, if $u \leq v$ and $u \neq v$. Finally, we say that $x \in \mathbb{R}^n$ dominates $y \in \mathbb{R}^n$ w.r.t. F if $F(x) \lneqq F(y)$.

Ideally, we would like to find a point simultaneously minimizing all the objectives $f_1, \dots, f_m$; however, such a solution is very unlikely to exist; instead, we rely on the concept of Pareto optimality.

Definition 2.1. A point $\bar{x} \in \Omega$ is *Pareto optimal* for problem (1) if there does not exist $y \in \Omega$ such that $F(y) \lneqq F(\bar{x})$. If there exists a neighborhood $\mathcal{N}(\bar{x})$ such that the previous property holds in $\Omega \cap \mathcal{N}(\bar{x})$, then $\bar{x}$ is *locally Pareto optimal*.

Pareto optimality is a strong property which is hard to attain in practice. A slightly weaker, but certainly more viable to obtain condition is weak Pareto optimality.

Definition 2.2. A point $\bar{x} \in \Omega$ is *weakly Pareto optimal* for problem (1) if there does not exist $y \in \Omega$ such that $F(y) < F(\bar{x})$. If there exists a neighborhood $\mathcal{N}(\bar{x})$ such that the previous property holds in $\Omega \cap \mathcal{N}(\bar{x})$, then $\bar{x}$ is *locally weakly Pareto optimal*.

The set of all Pareto optimal solutions constitutes the *Pareto set* of the problem. The image of the Pareto set through F is referred to as the *Pareto front*. We can now turn to the first order necessary conditions for Pareto optimality.

Definition 2.3. A point $\bar{x} \in \Omega$ is *Pareto-stationary* for problem (1) if, for all feasible directions $d \in D(\bar{x}) = \{v \in \mathbb{R}^n \mid \exists \bar{t} > 0 : \bar{x} + tv \in \Omega\ \forall\, t \in [0, \bar{t}\,]\}$, it holds

$$\max_{j=1,\dots,m} \nabla f_j(\bar{x})^T d \geq 0.$$

Under differentiability assumptions, Pareto-stationarity is a necessary condition for all kinds of Pareto optimality; note that the Pareto-stationarity condition can be compactly written as

$$\min_{d\in D(\bar{x})} \max_{j=1,\dots,m} \nabla f_j\big(\bar{x}\big)^T d = 0.$$

Now, let us address well known results for unconstrained problems of the form

$$\min_{x\in\mathbb{R}^n} F(x) = \big(f_1(x), \dots, f_m(x)\big)^T. \tag{2}$$

Pareto optimality notions match those of the constrained case. Even Pareto-stationarity can be defined as in Definition 2.3, recalling that in such case $D(x) = \mathbb{R}^n$ for all $x \in \mathbb{R}^n$.

If a point $\bar{x}$ is not a Pareto-stationary point for problem (2), then there exists a direction which is a descent direction w.r.t. all objective functions. Hence (according to Fliege and Svaiter, 2000, sec. 3.1) we can define the steepest common descent direction as the solution of problem

$$\min_{\substack{d\in\mathbb{R}^n \\ \|d\|\leq 1}} \max_{j=1,\dots,m} \nabla f_j(\bar{x})^T d, \tag{3}$$

which, if ℓ_∞ norm is employed, can be reformulated as the LP problem

$$\begin{aligned} \min_{\beta\in\mathbb{R},\, d\in\mathbb{R}^n} \quad & \beta \\ \text{s.t.} \quad & -1 \leq d_i \leq 1 \quad \forall\, i = 1, \dots, n, \\ & \nabla f_j(\bar{x})^T d \leq \beta \quad \forall\, j = 1, \dots, m. \end{aligned}$$

Note that a slightly different characterization of steepest common descent directions, based on an ℓ_2-regularized formulation of problem (3) and again proposed by Fliege and Svaiter (2000), could be employed. Here we preferred to use formulation (3) because of the simplicity of the LP problem.

The solution of problem (3) may in fact be not unique, but this is not a real technical issue; we can define function $\theta : \mathbb{R}^n \to \mathbb{R}$ such that $\theta(\bar{x})$ indicates the optimal value of problem (3) at $\bar{x}$; function θ is continuous. We also denote by $v(\bar{x})$ the set of optimal solutions of (3), which is certainly nonempty. As previously stated, if $\bar{x}$ is Pareto-stationary, $\theta(\bar{x}) = 0$, if it is not, $\theta(\bar{x}) < 0$.

Now, based on the concept of steepest common descent, the standard (single-point) multi-objective steepest descent (MOSD) algorithm was proposed by Fliege and Svaiter (2000). We report the algorithm in Algorithm 1 .

The algorithm makes use of a backtracking Armijo-type line search, which is described in Algorithm 2. The idea of the latter procedure is

Algorithm 1: MultiObjectiveSteepestDescent.

Input: $F : \mathbb{R}^n \to \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$
$k = 0$
while x^k *is not Pareto stationary* **do**
  Compute
$$d^k \in \arg\min_{\substack{d\in\mathbb{R}^n \\ \|d\|\leq 1}} \max_{j=1,\dots,m} \nabla f_j(x^k)^T d$$
  $\alpha_k = \texttt{ArmijoTypeLineSearch}(F(\cdot), x^k, d^k)$
  $x^{k+1} = x^k + \alpha_k d^k$
  $k = k + 1$
return x^k

Algorithm 2: ArmijoTypeLineSearch.

Input: $F : \mathbb{R}^n \to \mathbb{R}^m$, $x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$, $\alpha_0 > 0$, $\delta \in (0,1)$, $\beta \in (0,1)$
$\alpha = \alpha_0$
while $F(x + \alpha d) \nleq F(x) + \beta\alpha J_F(x)d$ **do**
  $\alpha = \delta\alpha$
return α

that of reducing the step size as long as a sufficient decrease has not been reached for all the objective functions.

We now recall the main theoretical property characterizing the line search (Fliege and Svaiter, 2000, Lemma 4).

Lemma 2.1. *If F is continuously differentiable and $J_F(x)d < 0$ (i.e., $\theta(x) < 0$), then there exists some $\varepsilon > 0$, which may depend on x, d and β, such that*

$$F(x + td) < F(x) + \beta t J_F(x)d$$

for all $t \in (0, \varepsilon]$.

The above lemma guarantees finite termination of the line search procedure along a common descent direction. The following convergence properties (Fliege and Svaiter, 2000, Theorem 1 and Section 9.1) hold instead for the MOSD procedure.

Lemma 2.2. *Every accumulation point of the sequence $\{x^k\}$ produced by Algorithm 1 is a Pareto stationary point. If the function F has bounded level sets, in the sense that $\{x \in \mathbb{R}^n \mid F(x) \leq F(x^0)\}$ is bounded, then the sequence $\{x^k\}$ stays bounded and has at least one accumulation point.*

Now, we need to introduce relaxations to the concepts of Pareto-stationary and common descent directions. First, we recall the concept of ε-Pareto-stationarity introduced by Cocchi and Lapucci (2020).

Definition 2.4. Let $\varepsilon \geq 0$. A point $\bar{x} \in \mathbb{R}^n$ is ε-Pareto-stationary for problem (2) if

$$\min_{\substack{d\in\mathbb{R}^n \\ \|d\|\leq 1}} \max_{j=1,\dots,m} \nabla f_j(\bar{x})^T d \geq -\varepsilon.$$

Next, inspired by Cocchi et al. (2020), we can introduce the concept of *steepest partial descent* at $\bar{x}$ w.r.t. a subset of indices of objectives $I \subseteq \{1,\dots,m\}$. Given problem

$$\min_{\substack{d\in\mathbb{R}^n \\ \|d\|\leq 1}} \max_{j\in I} \nabla f_j(\bar{x})^T d,$$

we denote by $\theta^I(\bar{x})$ its optimal value and by $v^I(\bar{x})$ the set of optimal solutions, which we refer to as steepest partial descent directions w.r.t. I. Partial descent directions, if used appropriately, can be useful in algorithms to perform exploration steps to enrich the current Pareto set approximation. It is easy to see (by analogous reasonings as Cocchi et al., 2020, Proposition 3) that, if $\bar{x}$ is not a Pareto-stationary point for (2), then $\theta^I(\bar{x}) < 0$ for any $I \subseteq \{1,\dots,m\}$.

Finally, we recall, from Cocchi and Lapucci (2020), the definition of multi-objective augmented Lagrangian for problems with inequality constraints.

Definition 2.5. The *multi-objective augmented Lagrangian* function of penalty parameter τ associated with problem (1) is given by

$$\mathcal{L}_\tau(x,\mu) = F(x) + \frac{\tau}{2}\left(\sum_{i=1}^{p}\left(\max\left\{0, g_i(x) + \frac{\mu_i}{\tau}\right\}\right)^2\right)e,$$

where $\mu \geq 0 \in \mathbb{R}^p$ is the vector of Lagrange multipliers.

3. The algorithm

In this section, we describe the multiple-points multi-objective augmented Lagrangian method proposed in this paper, which we call FRONT-ALAMO, to solve problem (1). The algorithmic scheme is reported in Algorithm 3. Note that we have denoted by $\texttt{MultiobjectiveSteepestDescent}(\cdot,\cdot,\varepsilon_k)$ the procedure in Algorithm 1 run until the solution is ε_k-Pareto-stationary. We also denote by $\mathcal{L}^I_\tau(x,\mu)$ the components of $\mathcal{L}_\tau(x,\mu)$ indexed by I, and by θ_k and v_k maps θ and v associated with $\mathcal{L}_{\tau_k}(x,\mu^k)$.

Algorithm 3: FRONT-ALAMO.

Input: $\mu^0 \in \mathbb{R}^p_+$, $\bar{\mu} \geq 0$, $\rho > 1$, $\sigma \in (0,1)$, $\tau_0 > 0$, X^0 a list of feasible non-dominated points for the original problem, $\{\varepsilon_k\} \subset \mathbb{R}$ a decreasing sequence
for $k = 0, 1, \dots$ **do**
  Let $\mathcal{L}_{\tau_k}$ the current Augmented Lagrangian function defined as:
$$\mathcal{L}_{\tau_k}(x,\mu^k) = F(x) + \frac{\tau_k}{2}\left(\sum_{i=1}^{p}\left(\max\left\{0, g_i(x) + \frac{\mu_i^k}{\tau_k}\right\}\right)^2\right)e$$
  set $\hat{X}^k = X^k \setminus \{x \in X^k \mid \exists\, y \in X^k \text{ s.t. } \mathcal{L}_{\tau_k}(y,\mu^k) \lneq \mathcal{L}_{\tau_k}(x,\mu^k)\}$
  set $X_{\text{tmp}} = \hat{X}^k$
  for $x_c \in \hat{X}^k$ **do**
    for $I \in 2^{\{1,\dots,m\}}$ **do**
      if $\theta^I_k(x_c) < 0$ **then**
        set $d \in v^I_k(x_c)$
        set $\alpha = \texttt{ArmijoTypeLineSearch}(\mathcal{L}^I_{\tau_k}(\cdot,\mu^k), x_c, d)$
        set $z = \texttt{MultiObjectiveSteepestDescent}(\mathcal{L}_{\tau_k}(\cdot,\mu^k), x_c + \alpha d, \varepsilon_k)$
        if $\nexists\, y \in X_{\text{tmp}} : \mathcal{L}_{\tau_k}(y,\mu^k) \lneq \mathcal{L}_{\tau_k}(z,\mu^k)$ **then**
          set $X_{\text{tmp}} = X_{\text{tmp}} \setminus \{x \in X_{\text{tmp}} \mid \mathcal{L}_{\tau_k}(z,\mu^k) \lneq \mathcal{L}_{\tau_k}(x,\mu^k)\} \cup \{z\}$
  set $X^{k+1} = X_{\text{tmp}}$
  for $i = 1, \dots, p$ **do**
    set $V_i^{k+1} = \min\left\{\min_{x\in X^{k+1}}\{-g_i(x)\}, \frac{\mu_i^k}{\tau_k}\right\}$
    set $\mu_i^{k+1} = \max\left\{0, \min\{\mu_i^k + \tau_k \max_{x\in X^{k+1}}\{g_i(x)\}, \bar{\mu}\}\right\}$
  if $(\|V^{k+1}\| > \sigma\|V^k\|)$ ***or*** $(\exists\, x^{k+1} \in X^{k+1} \text{ s.t. } g_i(x^{k+1}) < 0$ ***and*** $\mu_i^k + \tau_k g_i(x^{k+1}) > 0$ for some $i \in \{1,\dots,p\})$ **then**
    set $\tau_{k+1} = \rho\tau_k$
  else
    set $\tau_{k+1} = \tau_k$

Through the iterations, the algorithm produces a sequence of sets of points $\{X^k\}$, which approximate the Pareto set of the original problem with increasing accuracy. At each iteration, an augmented Lagrangian

function defined as in Definition 2.5 is considered, with penalty parameter τ_k and multipliers μ^k. At the beginning of the generic iteration k, all points that are dominated w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$ are filtered out of the set; we denote such filtered set by $\hat{X}^k$. Now, the following iterate is initialized as $\hat{X}^k$; then, each point $x_c \in \hat{X}^k$ is used as a starting point for exploration; in particular, for any possible subset $I \subseteq \{1, \dots, m\}$ the steepest partial descent direction is explored by an `ArmijoTypeLineSearch` restricted to the components of $\mathcal{L}^I_{\tau_k}(x, \mu^k)$, provided that a partial descent direction actually exists. After the line search step, the obtained point is refined by steepest descent on all the objectives up to ε_k-Pareto-stationarity. If it is then not dominated w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$ by any other point currently in the new iterate set, it is consequently added to such set, while all points that are dominated by it are removed.

Once all points in $\hat{X}^k$ are tested, the constructed set will constitute the next iterate X^{k+1}. The multipliers and the penalty parameter are updated similarly as in the scalar `ALM` with multipliers safeguarding (Kanzow and Steck, 2017), with one key adjustment: to evaluate how much a constraint is violated, the worst violation attained on that constraint by any point in X^{k+1} is considered. In addition, the second clause of the conditional statement at line 18 allows to avoid unfortunate cases where a point which is strictly feasible w.r.t. some constraint g_i is unnecessarily pushed to satisfy it with a larger margin.

Remark 3.1. At each iteration k, the set X^{k+1} is a list of mutually nondominated points w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$. As we will shortly see, maintaining a set of mutually nondominated points with respect to the augmented Lagrangian does not provide theoretical asymptotic properties. However, this has a remarkable impact from a computational point of view: it allows, especially at late iterations, to remove solutions that are too far from feasibility or that have bad values for all the objectives; in addition, in practice the algorithm will be run for a large enough number of iterations and then stopped; the solutions in the returned set are mutually nondominated w.r.t. the final augmented Lagrangian; because of this property, most of the points in the returned set that are "sufficiently feasible" are nondominated also w.r.t. the original problem.

In the next section, we will show in detail that Algorithm 3 is well defined and we will carefully address its convergence properties.

4. Convergence analysis

In this section, we provide a rigorous formal analysis of Algorithm 3 from a theoretical perspective. We first show that the procedure is actually well defined and then we state its asymptotic convergence properties. Before proceeding, we need to make a reasonable assumption.

Assumption 4.1. The objective function F has bounded level sets in the multi-objective sense, i.e., the set $\{x \in \mathbb{R}^n \mid F(x) \leq z\}$ is bounded for any $z \in \mathbb{R}^m$.

Concerning algorithm well-definiteness, we begin by noting that the line search procedure at line 10 of Algorithm 3 stops in a finite time, producing a valid stepsize. Indeed, this result holds straightforwardly from Lemma 2.1 and the fact that the procedure is performed considering $\mathcal{L}^I_{\tau_k}(x, \mu^k)$, starting at a point x_c such that $\theta^I_k(x_c) < 0$.

The other nontrivial instruction of the `FRONT-ALAMO` procedure is step 11, that we address in the following proposition.

Proposition 4.1. *The `MultiObjectiveSteepestDescent` procedure at line 11 of Algorithm 3 stops in a finite number of iterations.*

Proof. Finite termination can be proved as in Proposition 4 from Cocchi and Lapucci (2020), recalling that Assumption 4.1 holds. □

Now, we are able to characterize the points belonging to each iterate set X^k.

Proposition 4.2. *Let $\{X^{k+1}\}$ be the sequence of sets generated by Algorithm 3. Then, for each k and for each $x^{k+1} \in X^{k+1}$, we have:*

(a) *x^{k+1} is not dominated by any other point in X^{k+1} w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$, i.e., there does not exist $y \in X^{k+1}$ such that $\mathcal{L}_{\tau_k}(y, \mu^k) \lneqq \mathcal{L}_{\tau_k}(x^{k+1}, \mu^k)$;*
(b) *x^{k+1} is ε_k-Pareto-stationary w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$.*

Proof. We prove the two statements one at a time:

(a) X^{k+1} is equal to X_{tmp} at the end of the main loop of each iteration, at step 14. X_{tmp} is initialized with $\hat{X}^k$, which contains mutually nondominated points w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$ by its definition at line 4. Then, X_{tmp} can be modified only at step 13, where a point is added only if it is nondominated, from the condition at line 12, and all points dominated by it are removed.

(b) We have two possible cases: $x^{k+1} \in \hat{X}^k$ or $x^{k+1} \notin \hat{X}^k$. In the latter case, x^{k+1} has necessarily been added to X_{tmp} through instructions 9–13; in particular, x^{k+1} was produced by instruction 11 and is thus ε_k-Pareto-stationary.
So, let us assume that $x^{k+1} \in \hat{X}^k$ and, by contradiction, that $\theta_k(x^{k+1}) < -\varepsilon_k$. In this case, $x_c = x^{k+1}$ would satisfy the conditions at step 8 for $I = \{1, \dots, m\}$, as $\theta^I_k(x^{k+1}) = \theta_k(x^{k+1}) < -\varepsilon_k < 0$. The line search hence is guaranteed, by Lemma 2.1, to find a step α such that $\mathcal{L}_{\tau_k}(x_c + \alpha d, \mu^k) < \mathcal{L}_{\tau_k}(x_c, \mu^k)$, and by the properties of the `MOSD` procedure we have $\mathcal{L}_{\tau_k}(z, \mu^k) \leq \mathcal{L}_{\tau_k}(x_c + \alpha d, \mu^k)$. Hence, this new point z (strictly) dominates x^{k+1} w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$. Now, from the instructions of the algorithm, either z belongs to X^{k+1} or there exists $y \in X^{k+1}$ such that $\mathcal{L}_{\tau_k}(y, \mu^k) \lneqq \mathcal{L}_{\tau_k}(z, \mu^k) < \mathcal{L}_{\tau_k}(x^{k+1}, \mu^k)$. However, this is absurd, since $x^{k+1} \in X^{k+1}$ and from statement (a) X^{k+1} contains mutually nondominated points. Hence, $\theta_k(x^{k+1}) \geq -\varepsilon_k$.

□

Let $\{X^k\}$ be the sequence of (finite) sets produced by the algorithm. In order to assess the asymptotic convergence properties of Algorithm 3, we need to consider sequences of points $\{x^k\}$ such that $x^k \in X^k$ for all k.

We are now able to begin the convergence analysis with a technical Lemma.

Lemma 4.3. *Let $\{X^k\}$ be the sequence of sets generated by Algorithm 3, and let $\{x^k\}$ be any sequence of points such that $x^k \in X^k$ for all k. Let $\bar{x}$ be a limit point of $\{x^k\}$, i.e., there exists an infinite subset $K \subseteq \{0, 1, \dots\}$ such that*

$$\lim_{\substack{k \to \infty \\ k \in K}} x^k = \bar{x},$$

and suppose that $g(\bar{x}) \leq 0$, i.e., $\bar{x} \in \Omega$. Then, for all $i = 1, \dots, p$ such that $g_i(\bar{x}) < 0$ we have

$$\max\{0, \mu^k_i + \tau_k g_i(x^{k+1})\} = 0$$

for all $k \in K$ sufficiently large.

Proof. Let $g_i(\bar{x}) < 0$ and $k_1 \in K$ be such that $g_i(x^{k+1}) < c < 0$ for all $k \geq k_1$, $k \in K$. From the instructions of the algorithm we know that $\mu^k_i \geq 0$ for all k. There are two possible cases:

(a) $\tau_k \to \infty$
The sequence $\{\mu^k\}$ is bounded by definition, hence there exists $k_2 \geq k_1$, $k_2 \in K$, such that for all $k \in K$, $k \geq k_2$ we have $\mu^k_i + \tau_k g_i(x^{k+1}) < 0$ and thus $\max\{0, \mu^k_i + \tau_k g_i(x^{k+1})\} = 0$.

(b) $\{\tau_k\}$ is bounded.
From instruction 18 of the algorithm, there must exist $k_2 \geq k_1$ such that, for all $k \geq k_2$, condition

$$\forall\, x \in X^{k+1} \quad \mu^k_j + \tau_k g_j(x) \leq 0 \quad \forall j \in \{1, \dots, p\} \text{ s.t. } g_j(x) < 0$$

holds. Hence, for $k \geq k_2$, $k \in K$, we have $\mu^k_i + \tau_k g_i(x^{k+1}) \leq 0$. Thus, we have $\max\{0, \mu^k_i + \tau_k g_i(x^{k+1})\} = 0$ for $k \in K$ sufficiently large.

□

Next, we prove feasibility of limit points of all possible points sequences $\{x^k\}$ produced by the algorithm.

Proposition 4.4. *Let $\{X^k\}$ be the sequence of sets generated by Algorithm 3, with $\varepsilon_k \to 0$, and let $\{x^k\}$ be any sequence of points such that $x^k \in X^k$ for all k. Let $\bar{x}$ be a limit point of $\{x^k\}$. Then, $\bar{x}$ is feasible for problem (1), i.e., $g(\bar{x}) \leq 0$.*

Proof. Let $K \subseteq \{0, 1, \dots\}$ be an infinite subset such that

$$\lim_{\substack{k\to\infty \\ k\in K}} x^{k+1} = \bar{x}.$$

If the sequence $\{\tau_k\}$ is bounded, from the instructions of the algorithm there must exist k_1 such that, for all $k > k_1$, we have $\|V^{k+1}\| \leq \sigma\|V^k\|$. Since $\sigma < 1$, this implies

$$\lim_{k\to\infty} \|V^k\| = 0,$$

i.e., for all $i \in \{1, \dots, p\}$,

$$\lim_{k\to\infty} V_i^{k+1} = \lim_{k\to\infty} \min\left\{ \min_{x\in X^{k+1}} \{-g_i(x)\}, \frac{\mu_i^k}{\tau_k} \right\} = 0.$$

Since by definition $\mu_i^k \geq 0$ for all i and k, it has to be

$$\lim_{k\to\infty} \min_{x\in X^{k+1}} \left\{-g_i(x)\right\} \geq 0.$$

But $\min_{x\in X^{k+1}} \{-g_i(x)\} \leq -g_i(x^{k+1})$. Hence

$$g_i(\bar{x}) = \lim_{\substack{k\to\infty \\ k\in K}} g_i\left(x^{k+1}\right) \leq \lim_{\substack{k\to\infty \\ k\in K}} \max_{x\in X^{k+1}} \left\{-g_i(x)\right\} \leq 0.$$

Now, assume $\tau_k \to \infty$. From Proposition 4.2, we know that each point $x \in X^{k+1}$ is ε_k-Pareto-stationary w.r.t. $\mathcal{L}_{\tau_k}(x, \mu^k)$. Hence

$$\max_{j=1,\dots,m} \left\{ \left(\nabla f_j(x^{k+1}) + \tau_k \sum_{i=1}^{p} \max\left\{0, g_i(x^{k+1}) + \frac{\mu_i^k}{\tau_k}\right\} \nabla g_i(x^{k+1}) \right)^T d \right\} \geq -\varepsilon_k \quad \forall\, d \in \mathbb{R}^n : \|d\| \leq 1.$$

Dividing both sides of the inequality by τ_k and taking the limits for $k \to \infty$, $k \in K$, recalling the continuity of J_F and J_g, the boundedness of d and $\{\mu^k\}$ and that $\tau_k \to \infty$, we get

$$\max_{j=1,\dots,m} \left\{ \left(\sum_{i=1}^{p} \max\{0, g_i(\bar{x})\} \nabla g_i(\bar{x}) \right)^T d \right\} \geq 0 \quad \forall\, d \in \mathbb{R}^n : \|d\| \leq 1,$$

which, since the arguments of the outer max operator are independent of j, is equal to

$$\frac{1}{2} \nabla \left(\| \max\{0, g(\bar{x})\} \|^2 \right)^T d \geq 0 \quad \forall\, d \in \mathbb{R}^n : \|d\| \leq 1,$$

where the max operator is intended component-wise. Thus, $\bar{x}$ is a critical point for problem

$$\min_{x\in\mathbb{R}^n} \frac{1}{2} \| \max\{0, g(x)\} \|^2.$$

Since $\Omega \neq 0$ and the above problem is convex, $\bar{x}$ is a global minimum point with $\max\{0, g(\bar{x})\} = 0$, i.e., $g(\bar{x}) \leq 0$. □

Finally, we show that limit points are Pareto-stationary for the original problem.

Proposition 4.5. *Let $\{X^k\}$ be the sequence of sets generated by Algorithm 3, with $\varepsilon_k \to 0$, and let $\{x^k\}$ be any sequence of points such that $x^k \in X^k$ for all k. Let $\bar{x}$ be a limit point of $\{x^k\}$. Then, $\bar{x}$ is Pareto-stationary for problem (1).*

Proof. Recalling that, from Proposition 4.2, x^{k+1} is ε_k-Pareto-stationary for $\mathcal{L}_{\tau_k}(x, \mu^k)$, the result follows as in Proposition 6 from Cocchi and Lapucci (2020), where Lemma 4.3 can be used in place of Lemma 9 from the referenced paper. □

Remark 4.1. Pareto-stationarity, which we are able to prove for limit points of FRONT-ALAMO, is the same property that holds for limit points of the sequence produced by the single point ALAMO and analogous, in the scalar context, to stationarity attained by limit points of scalar ALM. Therefore, it is reasonable to assume that stronger properties are unlikely to be obtained by an ALM-like algorithm.

Remark 4.2. In the literature of Pareto front constructing descent methods (Cocchi et al., 2020; Liuzzi et al., 2016), convergence analysis is based on the concept of linked sequence. A sequence $\{x^k\}$ is a linked sequence if, for all k, $x^k \in X^k$ and x^k is generated at iteration $k-1$ starting the search procedure from x^{k-1}. It is easy to see that linked sequences are a particular instance of the sequences of points considered in Propositions 4.4 and 4.5, hence the convergence result obtained for Algorithm 3 is somewhat stronger than those based on linked sequences.

Remark 4.3. In our theoretical analysis we assumed the existence of a limit point $\bar{x}$. As commonly done in the literature of augmented Lagrangian methods (Birgin and Martinez, 2014; Cocchi and Lapucci, 2020), we do not directly address properties of existence of limit points, leaving it to boundedness arguments on the sequences, level sets, lower-level feasible sets or restart strategies.

Remark 4.4. The SQP algorithm from Fliege and Vaz (2016) which is, to the best of our knowledge, the only other derivative-based method in the literature to generate an approximation of the Pareto front, has similar convergence properties as Algorithm 3, in the sense that limit points of sequences of solutions are Pareto-stationary. However, the setting is basically different, as the exploration phase of the SQP method is eventually stopped and all the obtained points are then independently driven to Pareto-stationarity by an iterative method. Convergence hence follows from a single-point mechanism. On the other hand, in Algorithm 3 exploration and convergence are performed somewhat in parallel, in an effectively multiple-points fashion.

5. Computational experiments

In this Section, we show the results of thorough computational experiments, focusing on the comparisons between FRONT-ALAMO and some state-of-the-art methods in the multi-objective constrained optimization context. All the tests were run on a computer with the following characteristics: Intel Xeon Processor E5-2430 v2 6 cores 2.50 GHz, 16 GB RAM. The code for all the algorithms considered in the experiments was written in Python3.

5.1. Experiment Settings

Before commenting the results, we list the state-of-the-art methods used in the comparisons with FRONT-ALAMO. In addition, we describe the tested problems and the metrics used in the comparisons.

5.1.1. Metrics

In order to evaluate the performance of the algorithms, we employed the three metrics defined by Custódio et al. (2011), which are very popular and used by the multi-objective optimization community: *purity*, Γ*–spread* and Δ*–spread*. We recall that the *purity* metric measures the quality of the generated front, that is, how good the non-dominated points computed by a solver are with respect to those obtained by the other ones. Clearly, a higher value is associated with a better performance. In order to calculate this metric, we need a reference front to which compare the generated front of an algorithm. In our experiments, the reference front was the one obtained by combining the fronts of all the considered algorithms and by discarding the dominated points. The *spread* metrics are equally essential because they measure the uniformity of the generated front in the objectives space. In particular, the Γ*–spread* is defined as the maximum ℓ_∞ distance in the objectives space between adjacent points of the Pareto front, and the Δ*–spread* is quite similar

to the standard deviation of the ℓ_∞ distances between adjacent Pareto front points. In these metrics, good performance is associated with a low value.

In addition to the three previous metrics, we evaluated the number of non-dominated points obtained by a method with respect to the reference front (*ND–points*). We supposed that this metric was as important as the *purity* one. Indeed, *ND–points* metric allowed us to see how many non-dominated points an algorithm was capable to obtain with respect to the whole reference front.

Lastly, we employed the popular performance profiles introduced by Dolan and Moré (2002), that are an useful tool to better appreciate the relative performance and robustness of the algorithms. The performance profile for a solver is the (cumulative) distribution function for the ratio of the value of the performance measure obtained by the solver to the best one of all of the solvers. In particular, it is the probability for solver s that one of its performance measure values achieved in a problem is within a factor $\tau \in \mathbb{R}$ of the best possible value obtained by all of the solvers in that problem. For a more detailed explanation about performance profiles, we refer to Dolan and Moré (2002). Note that performance profiles w.r.t. *purity* and *ND–points* were generated considering the inverse of the obtained values, since the metrics have increasing values for better solutions.

5.1.2. Algorithms and hyper-parameters

The choices about the FRONT-ALAMO hyper-parameters values were made based on some preliminary results on a subset of the tested problems, which we do not report here for the sake of brevity. The values are the following:

- $\tau_0 = 1$;
- if the problem only has bound constraints $\rho = 10$, otherwise $\rho = 2$;
- $\sigma = 0.9$;
- $\overline{\mu} = 10^4$;
- $\mu^0 = 0 \in \mathbb{R}^p$;
- in the ArmijoTypeLineSearch $\beta = 10^{-4}$ and $\delta = 0.5$.

The first algorithm we chose to use in the comparison with FRONT-ALAMO is MOSQP (Fliege and Vaz, 2016), which is a gradient-based method for constrained and unconstrained nonlinear multi-objective optimization problems that implements an SQP–type approach. Since it is the only gradient-based algorithm from the literature designed to produce Pareto front approximations, we consider it our most important competitor. The chosen hyper-parameters for MOSQP were the best ones according to Fliege and Vaz (2016). For the quadratic approximations, we used $H_i = I_m$ (I_m being the identity matrix) in the second stage and $H_i = \nabla^2 f_i(x^k) + E_i$ (E_i being obtained by a modified Cholesky algorithm) in the third stage, as Fliege and Vaz (2016) state that it is the most robust and efficient way to use MOSQP. For a more detailed explanation about the various MOSQP stages and versions, we refer to Fliege and Vaz (2016). Lastly, we used the Ipopt software package (Wächter and Biegler, 2006) (https://github.com/coin-or/Ipopt) in order to solve the SQP problems.

DMS (Custódio et al., 2011) is the second algorithm used in the comparisons. It is a multi-objective derivative-free methodology, which is inspired by the search/poll paradigm of direct-search methods of directional type and uses the concept of Pareto dominance to maintain a list of non-dominated points, from which the new iterates or poll centers are chosen. The hyper-parameters for DMS were set according to Custódio et al. (2011) and to the authors code (http://www.mat.uc.pt/dms).

The third and last algorithm is NSGA-II (Deb et al., 2002), which is a non-dominated sorting-based multi-objective evolutionary algorithm. In particular, NSGA-II is a genetic algorithm that is mainly composed by a fast non-dominated sorting approach and a selection operator that creates a mating pool by combining the parent and offspring populations and selecting the best N solutions. In contrast to the other algorithms, NSGA-II considers a fixed number of solutions in the pool, which was set to 100 in our experiments. The hyper-parameters of the algorithm were the ones chosen by Deb et al. (2002).

For each algorithm and problem, we decided to execute the test for up to 2 min. A termination criterion based on a time limit is the fairest way to evaluate the behavior of such diverse algorithms on the tested problems. Clearly, specific stopping criteria indicating that a certain algorithm cannot improve the solutions anymore were also taken into account. Since NSGA-II is the only non-deterministic algorithm used in the computational experiments, we decided to run it with 10 different seeds for the pseudo-random number generator. Every execution had the same time limit used for the other algorithms (2 min). After the execution of the 10 runs, we compared the fronts based on the *purity* metric and we chose the best one among them. In this case, the reference front for the comparison was obtained by combining the fronts of the 10 executions. We considered the resulting best front as NSGA-II output. Executing 10 runs lets NSGA-II reduce its sensibility with respect to the seed used for its random operations. Note however that, since we consider a best case scenario for NSGA-II, the overall comparison should be considered at least partially biased in favor of this algorithm. The other methods (FRONT-ALAMO, DMS, MOSQP) are deterministic. Therefore, they were executed once.

Table 1
Problems used in the computational experiments. #L.C. indicates the number of linear constraints (in this column the boundary constraints are not considered). #N.L.C. indicates the number of non linear constraints. B.C. indicates the type(s) of the boundary conditions.

PROBLEM	n	m	p	#L.C.	#N.L.C.	B.C.
M-BNH1	2	2	2	–	2	–
M-BNH2	2	2	2	–	2	–
LAP1	2	2	3	1	2	–
LAP2	2, 5, 10, 20, 30, 40, 50, 100, 200	2	1	–	1	–
M-OSY	6	2	18	4	2	lb, ub
CEC1, CEC2, CEC3, CEC7	5, 10, 20, 30, 40, 50, 100, 200	2	$2n$	–	–	lb, ub
CEC8, CEC9, CEC10	5, 10, 20, 30, 40, 50, 100, 200	3	$2n$	–	–	lb, ub
ZDT1, ZDT2	2, 5, 10, 20, 30, 40, 50, 100, 200	2	$2n$	–	–	lb, ub
MOP1	1	2	2	–	–	lb, ub
MOP2	2, 5, 10, 20, 30, 40, 50, 100, 200	2	$2n$	–	–	lb, ub
MOP3	2	2	4	–	–	lb, ub

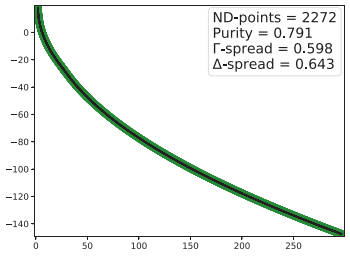
5.1.3. Problems

The tested problems are described in Table 1. In this benchmark, we considered problems whose objective functions are at least continuously differentiable almost everywhere. Since some problems present singularities, we counted them as Pareto-stationary points. All the constraints are defined by continuously differentiable convex functions.

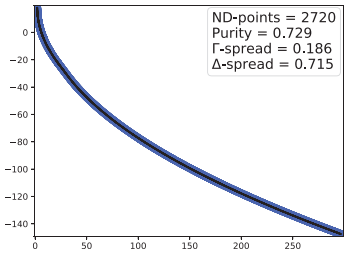
We included in our benchmark the slightly modified versions of the BNH problems and the LAP problems from Cocchi and Lapucci (2020). We also included a modification of the OSY problem (Osyczka and Kundu, 1995). The modified form of this problem can be found in Appendix A.

Furthermore, we included into the test set problems characterized only by boundary constraints: the CEC problems (Zhang et al., 2008), the ZDT problems (Zitzler et al., 2000) and the MOP problems (Huband et al., 2006). It is worth remarking that the CEC and ZDT problems have particularly difficult objective functions, so they are interesting to study the effectiveness of the algorithms when solving hard problems.

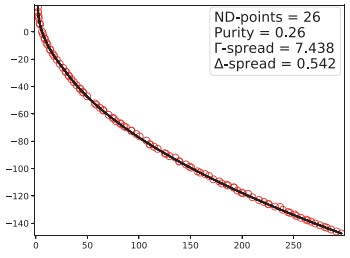
For each problem with general constraints, we started the algorithms from one feasible point (Table 2). In this way, we intended to study the exploration capabilities of the algorithms. Indeed, algorithms with great exploration abilities should create a spread and solid Pareto front on these problems. For the bound constrained problems (MOP1 is the only



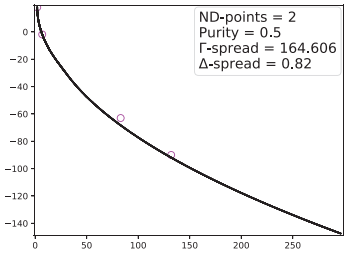
(a) FRONT-ALAMO Pareto front - M-BNH1



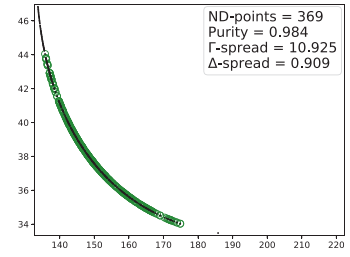
(b) DMS Pareto front - M-BNH1



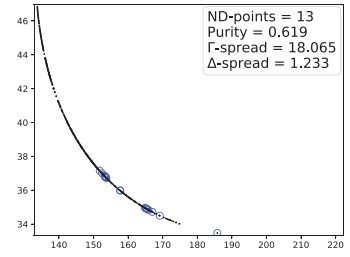
(c) NSGA-II Pareto front - M-BNH1



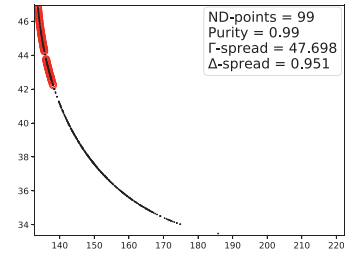
(d) MOSQP Pareto front - M-BNH1



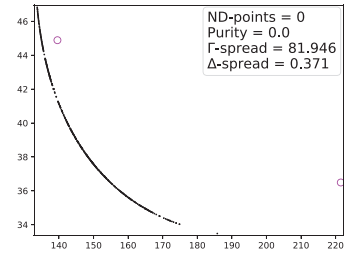
(e) FRONT-ALAMO Pareto front - M-BNH2



(f) DMS Pareto front - M-BNH2



(g) NSGA-II Pareto front - M-BNH2



(h) MOSQP Pareto front - M-BNH2

Fig. 1. Pareto front approximation for the four algorithms considering the M-BNH problems (For interpretation of the references to color in text, the reader is referred to the web version of the article).

Table 2
Initial points for the tested problems.

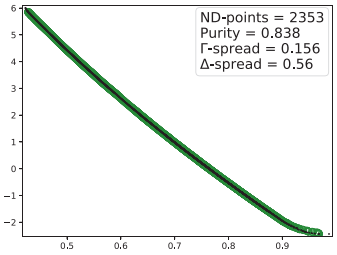
PROBLEM(S)	INITIAL POINT(S)
M-BNH1, LAP1, LAP2, MOP1	$0 \in \mathbb{R}^n$
M-BNH2	$[8, -3]$
M-OSY	$[2, 0, 1, 0, 1, 8]$
CEC, ZDT, MOP2, MOP3	Points from the hyper-diagonal

exception since it is too small in terms of number of dimensions), the initial points were uniformly selected from the hyper-diagonal defined by the lower and upper bounds, as done by Custódio et al. (2011). In these cases, the number of initial points is equal to the number of dimensions of the considered problem.

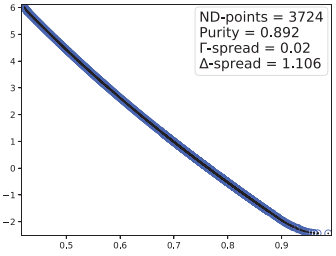
5.2. M-BNH, LAP1 and M-OSY problems

We begin by studying the performance of the considered algorithms on the M-BNH1, M-BNH2, LAP1 and M-OSY problems.

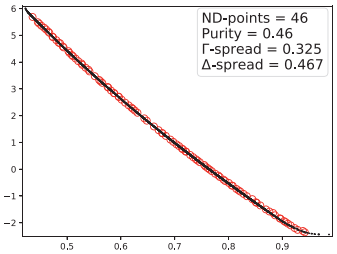
The results on the M-BNH problems (Fig. 1) show the great performance of FRONT-ALAMO with respect to the competitors: indeed, our method obtained the best *purity* value in the M-BNH1 problem and a *purity* value very close to the best one in the M-BNH2. In this latter problem, the differences with respect to our gradient-based competitor and DMS are even clearer, as FRONT-ALAMO obtained a *purity* value very close to 1. Here, the MOSQP method did not manage to obtain a single non-dominated point w.r.t. the competitors. Considering the Δ–*spread*, FRONT-ALAMO was the second best method in both problems. As for the Γ–*spread*, FRONT-ALAMO appears to have a decent behavior, being the second best algorithm in the M-BNH1 problem and outperforming all the competitors in the M-BNH2 problem. This behavior can



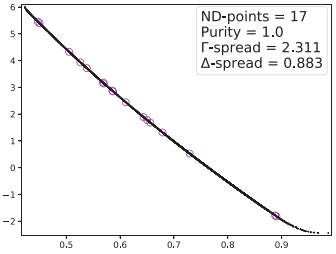
(a) FRONT-ALAMO Pareto front - LAP1



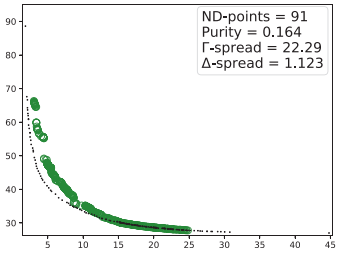
(b) DMS Pareto front - LAP1



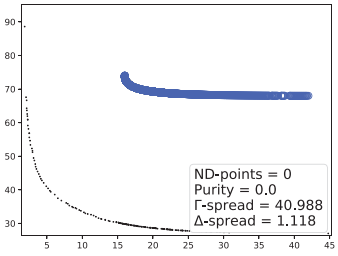
(c) NSGA-II Pareto front - LAP1



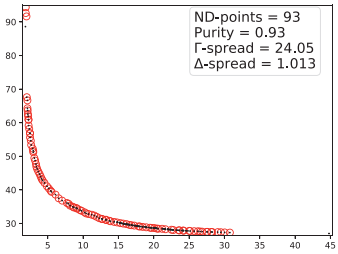
(d) MOSQP Pareto front - LAP1



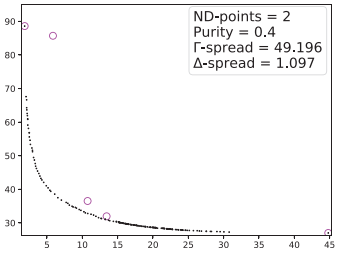
(e) FRONT-ALAMO Pareto front - M-OSY



(f) DMS Pareto front - M-OSY



(g) NSGA-II Pareto front - M-OSY



(h) MOSQP Pareto front - M-OSY

Fig. 2. Pareto front approximation for the four algorithms considering the LAP1 and M-OSY problems (For interpretation of the references to color in text, the reader is referred to the web version of the article).

also be observed on the LAP1 and M-OSY problems (Fig. 2). The worst algorithm turned out to be MOSQP, which seems to lack of search capabilities in the objectives space on the M-BNH problems. This fact can also be noted for the DMS and NSGA-II algorithms on the M-BNH2 problem, which turned out to be difficult to solve. DMS outperformed the NSGA-II method on the M-BNH1 problem in terms of *ND–points* and *purity*, while on the M-BNH2 one it is the opposite. Lastly, considering the *spread* metrics, DMS performed better in terms of Γ–*spread*, while NSGA-II outperformed DMS on the Δ–*spread*.

Considering the LAP1 problem (Fig. 2), FRONT-ALAMO performed very well and its scores are close to those of DMS, which was the overall best algorithm on this problem. This fact can be also seen in the front plots: the Pareto fronts obtained by the two algorithms are very similar, while the other two methods (NSGA-II and MOSQP) did not have the same performance. Among these two latter algorithms, the MOSQP method managed to outperform FRONT-ALAMO on one metric, that is the *purity*, while the NSGA-II algorithm performed better on the Δ–*spread*. In the M-OSY problem (Fig. 2) NSGA-II was the most effective obtaining an uniform and spread Pareto front. In this case, FRONT-ALAMO achieved some interesting results. First of all, it outperformed the DMS algorithm: this achievement is remarkable since DMS is gradient-free and can escape non optimal Pareto-stationary points, while our method is gradient-based. In addition, our algorithm obtained more non-dominated points and a better Γ–*spread* than its gradient-based competitor (MOSQP).

The last FRONT-ALAMO peculiarity that we can see from these plots is the number of non-dominated points it achieved. Only the DMS algorithm managed to obtain a much greater value of *ND–points* in some

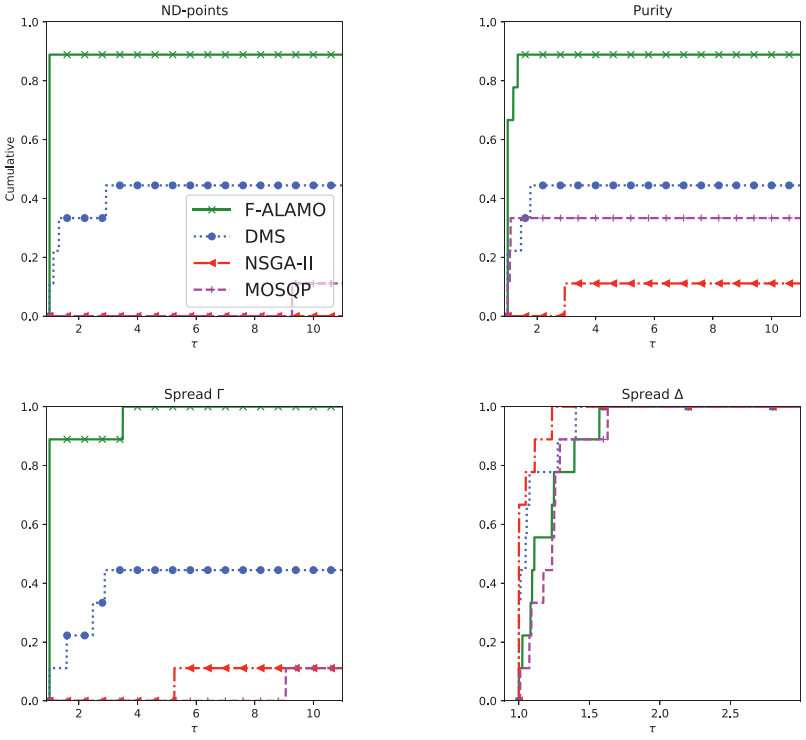
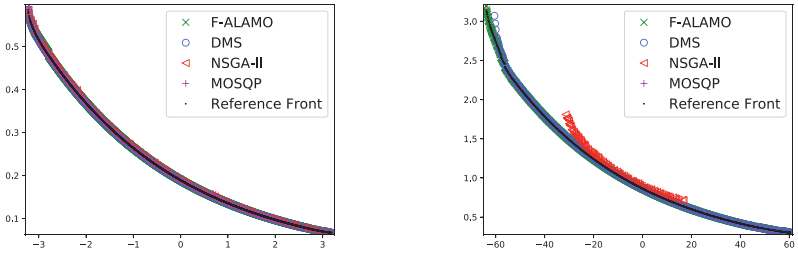
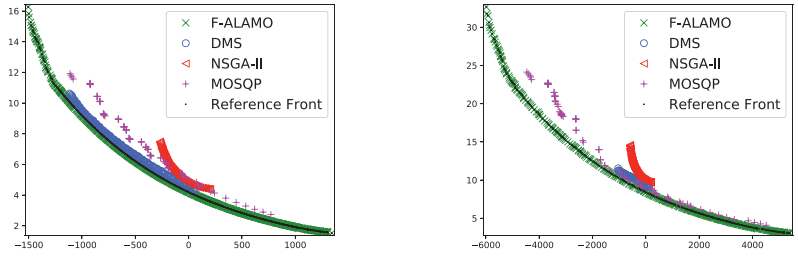


Fig. 3. Performance profiles for the four algorithms on the LAP2 problems. Each sub-figure represents the performance profiles considering as a performance measure one of the metrics explained in Section 5.1.1 (For interpretation of the references to color in text, the reader is referred to the web version of the article).



(a) LAP2, $n = 2$ (b) LAP2, $n = 10$



(c) LAP2, $n = 50$ (d) LAP2, $n = 100$

Fig. 4. Pareto front approximation for the four algorithms considering LAP2 problems at different dimensionalities (For interpretation of the references to color in text, the reader is referred to the web version of the article).

problems. However, in these problems FRONT-ALAMO was equally competitive in terms of this metric and the *purity* one.

5.3. LAP2 problems

The LAP2 problems represent another useful class of problems: indeed, they allow to discuss about the sensibility of the algorithms with regard to n, that is, how well they scale. Indeed, many algorithms have great performance considering small values of n. However, when a problem size grows, they lose their abilities to retrieve good Pareto front approximations. Before seeing some plots and metric values, we show the performance profiles considering all the LAP2 problems (Fig. 3).

The performance profiles highlight that FRONT-ALAMO strongly outperformed the other competitors with respect to *ND–points, purity* and Γ*–spread*. The second most robust algorithm is the DMS one. As for the Δ*–spread*, the methods performed similarly: here, it is very difficult to indicate the best algorithm. This fact can be a proof that no algorithm suffered from a non-uniformity in their fronts. On the contrary, in terms of the Γ*–spread* there is a clear winner: FRONT-ALAMO.

The motivation of these different results on the two *spread* metrics can be explained through the Fig. 4, where we show the Pareto fronts in four different LAP2 problems. Here, we show the fronts all together in order to provide a more direct impression of the results. Indeed, in

Table 3
Metrics values obtained by the four algorithms in the LAP2 problems with $n = 2, 10, 50, 100$. The values marked in bold are the best values (each of which is related to a specific score) obtained in a specific problem.

PROBLEM	METRIC	FRONT-ALAMO	DMS	NSGA-II	MOSQP
LAP2, $n = 2$	*ND–points*	**3680**	3256	34	397
	purity	0.747	**1.0**	0.34	0.936
	Γ–spread	0.123	**0.035**	0.185	0.556
	Δ–spread	0.722	0.547	**0.518**	0.844
LAP2, $n = 10$	*ND–points*	**1395**	477	0	23
	purity	0.775	0.633	0.0	**0.92**
	Γ–spread	**1.149**	3.309	43.472	27.501
	Δ–spread	0.997	1.031	**0.808**	0.999
LAP2, $n = 50$	*ND–points*	**944**	0	0	0
	purity	**1.0**	0.0	0.0	0.0
	Γ–spread	**39.861**	1275.832	1235.656	570.599
	Δ–spread	1.003	**0.927**	0.928	1.165
LAP2, $n = 100$	*ND–points*	**430**	0	0	0
	purity	**1.0**	0.0	0.0	0.0
	Γ–spread	**90.54**	5274.549	5384.548	1517.672
	Δ–spread	0.996	0.983	**0.972**	1.141

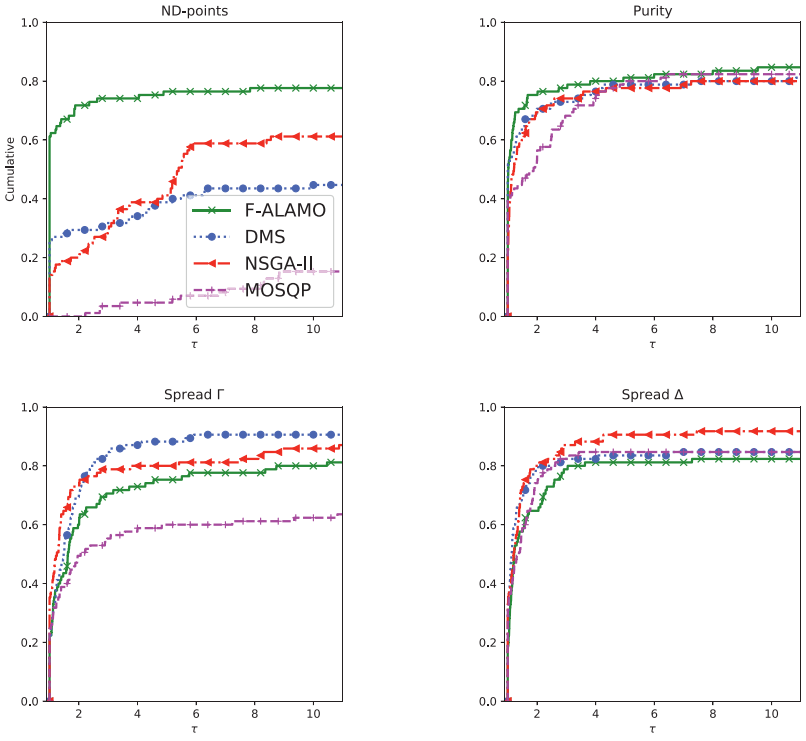


Fig. 5. Performance profiles for the four algorithms on the CEC, ZDT and MOP problems. Each sub-figure represents the performance profiles considering as a performance measure one of the metrics explained in Section 5.1.1 (For interpretation of the references to color in text, the reader is referred to the web version of the article).

the LAP2 problems, FRONT-ALAMO results show the superiority of our method at exploring the objectives space and creating a spread and uniform Pareto front. The competitors obtained large Γ–*spread* values with respect to those of FRONT-ALAMO, since they struggle to explore the extreme regions of the front.

When $n = 2$, all the methods managed to obtain the same Pareto front. However, increasing n, the differences between them become more and more clear. For instance, NSGA-II performance got worse with $n \geq 10$. It seems to be unable to spread the search in the objectives space and, also, to create a good, although small, Pareto front. The other gradient-free method (DMS) performed better but still it hardly reached the extremes of the objectives space. MOSQP seems not to have this last negative feature but it generally retrieved very few points and, in addition, most of them are dominated. However, the MOSQP performance was good with $n \leq 10$.

The above comments on the algorithm behaviors on the LAP2 problems are also supported by the numbers in Table 3. Observe that FRONT-ALAMO, whose performance was quite good on problems with small dimension, outperformed the competitors as the value of n increased. The superiority of our method when the dimension of a problem is high is very remarkable, especially considering *ND–points* (we obtained the best value, by far, for this metric in all the four problems), *purity* and Γ–*spread*. As we just highlighted commenting the performance profiles, all the algorithms performed well regarding the Δ–*spread* metric.

5.4. CEC, ZDT and MOP problems

In this last section of computational experiments, we comment the results on the problems characterized only by boundary constraints. Some of these are very difficult to solve and, in detail, the hardest ones are the CEC and ZDT problems. The CEC problems have non-continuously differentiable objective functions. The same feature is present in the ZDT problems, where the objective functions are also

composite. For the sake of brevity, we preferred to show the performance profiles related to all these problems. The performance profiles are shown in Fig. 5.

Regarding the *ND–points* metric, FRONT−ALAMO managed to outperform the competitors once again. Indeed, we can conclude that this is one of the most important peculiarities of our method: its capabilities allow it to expand the search towards a great portion of the objectives space and to retrieve many Pareto points. Considering this metric, the other two best competitors were NSGA−II and DMS: because of their gradient-free nature, allowing them to potentially escape from non optimal Pareto-stationary points, they managed to obtain good results in the most complex functions. On the contrary, the performance difference between FRONT−ALAMO and the other gradient-based method (MOSQP) is very sharp.

The performance profiles on the *purity* metric highlight the effectiveness of our method: it was not obvious, a priori, to obtain such great results with such complex functions, especially when some of our competitors are derivative-free.

Lastly, considering the *spread* metrics, our results are competitive with respect to the other competitors. In particular, in the Γ–*spread* performance profiles FRONT−ALAMO was the third best algorithm, while the gradient-free methods (NSGA−II and DMS) managed to have slightly better performance. Regarding the Δ–*spread*, NSGA−II turned out to be the most robust algorithm. However, the performance profiles on this metric are another proof of the effectiveness of the four algorithms to retrieve an uniform Pareto front.

6. Conclusions

In this paper, we considered smooth multi-objective optimization problems subject to convex constraints. We focused on the task of generating good Pareto front approximations for this class of problems. After a brief review of the existing literature, we proposed an Augmented Lagrangian Method specifically designed for this task.

The method represents an extension of the ALAMO procedure from Cocchi and Lapucci (2020), which is designed to produce a single Pareto-stationary solution. The proposed algorithm handles, at each iteration, a list of points that are mutually non-dominated and Pareto-stationary with respect to the current multi-objective augmented Lagrangian. Line searches along steepest common and partial descent directions are employed to carry out an exploration of the objectives space. The penalty parameter and the Lagrange multipliers are updated taking into account constraints violations committed by all the points in the current list.

For this algorithm, we proved global convergence to Pareto-stationarity of the sequences of points in the iterates lists. This type of convergence is more general than that based on linked sequences. With respect to the only other derivative-based method for this kind of problems, the SQP from (Fliege and Vaz, 2016), we obtain similar asymptotic properties for the limit points, but our method does not stop the exploration phase after a finite number of iterations.

Moreover, thorough computational experiments show that our method outperforms the SQP algorithm in terms of popular metrics for multi-objective optimization. We also compared the proposed procedure with the state-of-the-art derivative-free (DMS) and genetic (NSGA−II) approaches. Our procedure proved to obtain better results even w.r.t. the two mentioned ones.

Declaration of Competing Interest

The authors declare that they have no conflict of interest.

Acknowledgment

The authors would like to thank professor Marco Sciandrone for the precious discussions.

Appendix A. Modified form of the OSY problem

In this Appendix, we introduce the modified version of the OSY problem (Osyczka and Kundu, 1995) used in the computational experiments. The modification is carried out in order to make the feasible set and the objective functions convex.

$$\min_{x\in\mathbb{R}^6} \quad \begin{array}{l} f_1(x) = 25(x_1-2)^2 + (x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2 + (x_5-1)^2 \\ f_2(x) = \sum_{i=1}^{6} x_i^2 \end{array}$$

$$\text{s.t.} \quad \begin{array}{l} x_1 + x_2 - 2 \geq 0, \\ 6 - x_1 - x_2 \geq 0, \\ 2 - x_2 + x_1 \geq 0, \\ 2 - x_1 + 3x_2 \geq 0, \\ 4 - (x_3-3)^2 - x_4 \geq 0, \\ -(x_5-3)^2 + x_6 - 4 \geq 0, \\ 0 \leq x_1, x_2, x_6 \leq 10, \\ 1 \leq x_3, x_5 \leq 5, \\ 0 \leq x_4 \leq 6. \end{array}$$

References

Birgin, E.G., Martinez, J.M., 2014. Practical Augmented Lagrangian Methods for Constrained Optimization, 10. SIAM.
Campana, E.F., Diez, M., Liuzzi, G., Lucidi, S., Pellegrini, R., Piccialli, V., Rinaldi, F., Serani, A., 2018. A multi-objective DIRECT algorithm for ship hull optimization. Comput. Optim. Appl. 71 (1), 53–72.
Carrizosa, E., Frenk, J.B.G., 1998. Dominating sets for convex functions with some applications. J. Optim. Theory Appl. 96 (2), 281–295.
Cocchi, G., Lapucci, M., 2020. An augmented Lagrangian algorithm for multi-objective optimization. Comput. Optim. Appl. 77 (1), 29–56.
Cocchi, G., Liuzzi, G., Lucidi, S., Sciandrone, M., 2020. On the convergence of steepest descent methods for multiobjective optimization. Comput. Optim. Appl. 1–27.
Custódio, A.L., Madeira, J.F.A., Vaz, A.I.F., Vicente, L.N., 2011. Direct multisearch for multiobjective optimization. SIAM J. Optim. 21 (3), 1109–1140.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (2), 182–197.
Dolan, E.D., Moré, J.J., 2002. Benchmarking optimization software with performance profiles. Math. Program. 91 (2), 201–213.
Drummond, L.M.G., Iusem, A.N., 2004. A projected gradient method for vector optimization problems. Comput. Optim. Appl. 28 (1), 5–29.
Drummond, L.M.G., Maculan, N., Svaiter, B.F., 2008. On the choice of parameters for the weighting method in vector optimization. Math. Program. 111 (1-2), 201–216.
Eichfelder, G., 2009. An adaptive scalarization method in multiobjective optimization. SIAM J. Optim. 19 (4), 1694–1718.
Fliege, J., 2001. OLAF—A general modeling system to evaluate and optimize the location of an air polluting facility. OR-Spektrum 23 (1), 117–136.
Fliege, J., 2004. Gap-free computation of Pareto-points by quadratic scalarizations. Math. Methods Oper. Res. 59 (1), 69–89.
Fliege, J., Drummond, L.M.G., Svaiter, B.F., 2009. Newton's method for multiobjective optimization. SIAM J. Optim. 20 (2), 602–626.
Fliege, J., Svaiter, B.F., 2000. Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. 51 (3), 479–494.
Fliege, J., Vaz, A.I.F., 2016. A method for constrained multiobjective optimization based on SQP techniques. SIAM J. Optim. 26 (4), 2091–2119.
Fu, Y., Diwekar, U.M., 2004. An efficient sampling approach to multiobjective optimization. Ann. Oper. Res. 132 (1-4), 109–134.
Gass, S., Saaty, T., 1955. The computational algorithm for the parametric objective function. Naval Res. Logist. Q. 2 (1-2), 39–45.
Geoffrion, A.M., 1968. Proper efficiency and the theory of vector maximization. J. Math. Anal. Appl. 22 (3), 618–630.
Gravel, M., Martel, J.M., Nadeau, R., Price, W., Tremblay, R., 1992. A multicriterion view of optimal resource allocation in job-shop production. Eur. J. Oper. Res. 61 (1-2), 230–244.
Huband, S., Hingston, P., Barone, L., While, L., 2006. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans. Evol. Comput. 10 (5), 477–506. doi:10.1109/TEVC.2005.861417.
Jüschke, A., Jahn, J., Kirsch, A., 1997. A bicriterial optimization problem of antenna design. Comput. Optim. Appl. 7 (3), 261–276.
Kanzow, C., Steck, D., 2017. An example comparing the standard and safeguarded augmented Lagrangian methods. Oper. Res. Lett. 45 (6), 598–603.
Kasperska, R., Ostwald, M., Rodak, M., 2004. Bi-criteria optimization of open cross section of the thin-walled beams with flat flanges. In: PAMM: Proceedings in Applied Mathematics and Mechanics, 4. Wiley Online Library, pp. 614–615.
Konak, A., Coit, D.W., Smith, A.E., 2006. Multi-objective optimization using genetic algorithms: a tutorial. Reliab. Eng. Syst. Saf. 91 (9), 992–1007.
Laumanns, M., Thiele, L., Deb, K., Zitzler, E., 2002. Combining convergence and diversity in evolutionary multiobjective optimization. Evol. Comput. 10 (3), 263–282.
Leschine, T.M., Wallenius, H., Verdini, W.A., 1992. Interactive multiobjective analysis and assimilative capacity-based ocean disposal decisions. Eur. J. Oper. Res. 56 (2), 278–289.

Liuzzi, G., Lucidi, S., Parasiliti, F., Villani, M., 2003. Multiobjective optimization techniques for the design of induction motors. IEEE Trans. Magn. 39 (3), 1261–1264.

Liuzzi, G., Lucidi, S., Rinaldi, F., 2016. A derivative-free approach to constrained multiobjective nonsmooth optimization. SIAM J. Optim. 26 (4), 2744–2774.

Mostaghim, S., Branke, J., Schmeck, H., 2007. Multi-objective particle swarm optimization on computer grids. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. ACM, pp. 869–875.

Osyczka, A., Kundu, S., 1995. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Struct. Optim. 10 (2), 94–99. doi:10.1007/BF01743536.

Palermo, G., Silvano, C., Valsecchi, S., Zaccaria, V., 2003. A system-level methodology for fast multi-objective design space exploration. In: Proceedings of the 13th ACM Great Lakes Symposium on VLSI. ACM, pp. 92–95.

Pascoletti, A., Serafini, P., 1984. Scalarizing vector optimization problems. J. Optim. Theory Appl. 42 (4), 499–524.

Pellegrini, R., Campana, E.F., Diez, M., Serani, A., Rinaldi, F., Fasano, G., Iemma, U., Liuzzi, G., Lucidi, S., Stern, F., et al., 2014. Application of derivative-free multi-objective algorithms to reliability-based robust design optimization of a high-speed catamaran in real ocean environment1. In: Engineering Optimization IV-Rodrigues et al.(Eds.), p. 15.

Shan, S., Wang, G.G., 2005. An efficient Pareto set identification approach for multiobjective optimization on black-box functions. J. Mech. Des. 127 (5), 866–874.

Steuer, R.E., Choo, E.-U., 1983. An interactive weighted Tchebycheff procedure for multiple objective programming. Math. Program. 26 (3), 326–344.

Sun, Y., Ng, D.W.K., Zhu, J., Schober, R., 2016. Multi-objective optimization for robust power efficient and secure full-duplex wireless communication systems. IEEE Trans. Wirel. Commun. 15 (8), 5511–5526.

Tavana, M., 2004. A subjective assessment of alternative mission architectures for the human exploration of Mars at NASA using multicriteria decision making. Comput. Oper. Res. 31 (7), 1147–1164.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106 (1), 25–57.

White, D.J., 1998. Epsilon-dominating solutions in mean-variance portfolio analysis. Eur. J. Oper. Res. 105 (3), 457–466.

Zadeh, L., 1963. Optimality and non-scalar-valued performance criteria. IEEE Trans. Autom. Control 8 (1), 59–60.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S., 2008. Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report 264, 1–30.

Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. 8 (2), 173–195. doi:10.1162/106365600568202.