Computational Optimization and Applications (2024) 87:149–179
https://doi.org/10.1007/s10589-023-00510-2

A trust-region approach for computing Pareto fronts in multiobjective optimization

A. Mohammadi[1] · A. L. Custódio[2]

Received: 25 November 2022 / Accepted: 18 July 2023 / Published online: 20 August 2023

Abstract

Multiobjective optimization is a challenging scientific area, where the conflicting nature of the different objectives to be optimized changes the concept of problem solution, which is no longer a single point but a set of points, namely the Pareto front. In a posteriori preferences approach, when the decision maker is unable to rank objectives before the optimization, it is important to develop algorithms that generate approximations to the complete Pareto front of a multiobjective optimization problem, making clear the trade-offs between the different objectives. In this work, an algorithm based on a trust-region approach is proposed to approximate the set of Pareto critical points of a multiobjective optimization problem. Derivatives are assumed to be known, allowing the computation of Taylor models for the different objective function components, which will be minimized in two main steps: the extreme point step and the scalarization step. The goal of the extreme point step is to expand the approximation to the Pareto front, by moving towards the extreme points of it, corresponding to the individual minimization of each objective function component. The scalarization step attempts to reduce the gaps on the Pareto front, by solving adequate scalarization problems. The convergence of the method is analyzed and numerical experiments are reported, indicating the relevance of each feature included in the algorithmic structure and its competitiveness, by comparison against a state-of-art multiobjective optimization algorithm.

Keywords Multiobjective optimization · Trust-region methods · Taylor models · Scalarization techniques · Pareto front

✉ A. L. Custódio
alcustodio@fct.unl.pt

A. Mohammadi
a.mohammadi@campus.fct.unl.pt

[1] Center for Mathematics and Applications (NOVA MATH), NOVA School of Science and Technology, Campus de Caparica, 2829-516 Caparica, Portugal

[2] Department of Mathematics, Center for Mathematics and Applications (NOVA MATH), NOVA School of Science and Technology, Campus de Caparica, 2829-516 Caparica, Portugal

1 Introduction

Multiobjective optimization is an important field of research, strongly connected with applications. Examples of multiobjective optimization problems appear in different scientific domains such as engineering, environment, finance, or medicine [1, 2, 25, 29]. It is common to have more than one objective to optimize, being these objectives conflicting among each other. In this situation, the problem solution is no longer a single point, but it is now a set of points, for which it is not possible to improve one objective, without deteriorating the value of another, known as the Pareto front of the problem.

Typical approaches to this class of problems include the use of metaheuristics, like is the case of evolutionary algorithms, when we aim at an approximation to the complete Pareto front of the problem [12, 21], or aggregation techniques, that combine the different objectives into a single function, which is then optimized, generating a single point in the Pareto front of the problem [11, 20]. General theoretical properties have not yet been established for the class of methods based on metaheuristics. Regarding aggregation techniques, when addressing general nonconvex problems, if the goal is to compute an approximation to the complete Pareto front, care must be taken on the selection of an adequate scalarization technique, since the Pareto front cannot be simply retrieved with linear combinations of the corresponding objectives [7, 8]. Recently, some algorithms based on SQP techniques were successfully proposed [3, 15] for this task.

When the goal is just to compute a single Pareto point, many approaches can be found in the literature, that correspond to multiobjective variants of single objective algorithms [17]. These include Steepest Descent [14], Newton's Method [13], Quasi-Newton's approaches [23], among many others. Often, in the numerical experiments, algorithms are run from different initializations in an attempt of generating different points in the Pareto front. However, there is no guarantee of success and the algorithms do not incorporate any explicit mechanism for that purpose.

In the present work, we will follow a trust-region approach, based on quadratic Taylor models, to develop an algorithm that computes an approximation to the complete Pareto front of a multiobjective problem, by approximating the set of Pareto critical points. Trust-region methods have been extensively used in single objective optimization [5], leading to competitive implementations and commercial software [22]. The general framework of a trust-region based algorithm is relatively simple. At each iteration, a model of the objective function is computed and minimized in a trust-region, typically a ball around the current iterate, with a given radius. Depending on how well the model predicts the actual decrease in the objective function value, the model minimizer is accepted or rejected and the trust-region radius is updated.

The update rule for the trust-region radius is crucial for the success of the algorithm, both for single objective optimization or when simultaneously minimizing different models, corresponding to different components of the objective functions. Recently [28], encouraging numerical results have been provided regarding an updating strategy based on the hypervolume indicator [16]. Although, no theoretical results associated with the behavior of the algorithm were provided.

Trust-region algorithms with a well-established convergence analysis were already proposed for multiobjective optimization [4, 26, 27]. The original Newton's method [13], developed for convex unconstrained optimization, was extended in [4] to non-convex problems in a trust-region framework, by considering an additional set of linear inequality constraints that enforce descent. This set of additional constraints is also considered in [27], but now under a nonmonotone globalization technique. In [26], the additional constraints are not used but a strong local convexity assumption is needed. Quasi-Newton quadratic models are considered, where the Hessian matrix is positive definite. All algorithms presented in [4, 26, 27] are designed to obtain a single Pareto critical point for the multiobjective problem.

In our approach, we consider unconstrained general multiobjective optimization problems. The proposed algorithm will use quadratic Taylor models, not necessarily convex. No additional constraints are imposed. The algorithm incorporates an explicit strategy to compute an approximation to the complete Pareto front of the multiobjective problem, that relies on an extreme point step and a scalarization step. The extreme point step attempts to extend the approximation to the Pareto front by moving towards its extreme points, corresponding to the individual minimization of each function component. The scalarization step attempts to close the gaps in the Pareto front by selecting an adequate point and performing an aggregated minimization of the models corresponding to the different components of the objective function.

Section 2 formalizes the problem to be addressed and details the proposed algorithmic structure. Convergence is analyzed in Sect. 3. Section 4 provides numerical results that illustrate the relevance of each one of the proposed algorithmic features and compares the numerical performance of the method against the state-of-art Multi-objective Sequential Quadratic Programming (MOSQP) algorithm, proposed in [15]. Finally, some conclusions are presented in Sect. 5.

2 Algorithm description

In this work, we are interested in solving the multiobjective optimization problem defined as:

$$\begin{aligned} &\min F(x) \\ &\text{s.t. } x \in \mathbb{R}^n, \end{aligned} \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^q$, $F(x) = (f_1(x), \ldots, f_q(x))$, $n, q \in \mathbb{N}$, and $q \geq 2$. The objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, q$ are assumed to be twice continuously differentiable, with available gradients and Hessians, and conflicting among each other, meaning that it is not possible to find a single point that simultaneously minimizes all function components. The problem solution will be the so-called Pareto front of the problem, namely a set of nondominated points. In multiobjective optimization, we say that point x dominates point y, and represent it by $x \prec_F y$, if and only if $F(y) - F(x) \in \mathbb{R}^q_+ \backslash \{0\}$.

When trying to develop an algorithm that approximates the Pareto front of a multiobjective optimization problem, two goals should be taken into consideration: the extent of the Pareto front, meaning the capability of being able to compute the extreme points of it, corresponding to the individual minimization of each component of the objective function, and the density of the Pareto front, associated with the ability of the algorithm in fulfilling the gaps between points lying in the computed approximation to the Pareto front. The proposed Multiobjective Trust-Region (MOTR) algorithm addresses each one of these goals with two different steps, namely the *Extreme Point Step* and the *Scalarization Step*, which are performed at alternate iterations. The latter will make use of an additional step, the *Middle Point Step*, to select the points where scalarization problems will be solved. Algorithm 1 formalizes the main procedure.

Algorithm 1. MOTR

```
Input
    Initial list of nondominated points $L_0$, defined by (2).

For $k = 0, 1, 2, \ldots$
    If $mod(k, 2) = 0$, then go to the Extreme Point Step. Else go to the
    Scalarization Step.
    If some stopping criterion is met, then return.
End For
```

MOTR keeps a list of nondominated points and associated quantities, defined as

$$L = \left\{ (x^j, F(x^j), \Delta_{ep}^j, \Delta_{sc}^j) \mid j \in J \right\}, \tag{2}$$

where $J \subset \mathbb{N}$ is the set of indexes of the points in the list, Δ_{ep}^j is a $q \times 1$ vector storing at each component the trust-region radius associated with the point and the corresponding component of the objective function, to be used at the extreme point step, and $\Delta_{sc}^j \in \mathbb{R}^+$ represents the trust-region radius to be used at the scalarization step. Through this work, in a clear abuse of notation that facilitates reading, it will be often stated $x^j \in L$, meaning that $(x^j, F(x^j), \Delta_{ep}^j, \Delta_{sc}^j) \in L$.

In any of the two steps, points are selected from this list and quadratic Taylor models are built centered at the selected point, to replace the different components of the objective function. Thus, for $i = 1, \ldots, q$, the quadratic model m_i approximating f_i around a given point x_{step} is defined as

$$\begin{aligned} m_i(x) = {} & f_i(x_{step}) + \nabla f_i(x_{step})^\top (x - x_{step}) \\ & + \frac{1}{2}(x - x_{step})^\top \nabla^2 f_i(x_{step})(x - x_{step}), \end{aligned} \tag{3}$$

where $\nabla f_i(x_{step})$ and $\nabla^2 f_i(x_{step})$ represent the gradient vector and the Hessian matrix of f_i computed at x_{step}.

Depending on the step, models are used in different ways, to compute new nondominated points that are added to the list. Each time that a new point is added to the list, all dominated points are removed from it. Sections 2.1 and 2.2 detail the extreme point and the scalarization steps, respectively.

2.1 Extreme point step

The main goal of the extreme point step, described in Algorithm 2, is to expand the approximation to the Pareto front by moving towards the extreme points of it, corresponding to the individual minimization of each of the objective function components.

Algorithm 2. Extreme point step

```
Input
    Current list of nondominated points $L_k$, defined by (2).
    The minimum trust-region radius value $\Delta_{ep}^{min}$.
    Values for the parameters $0 < \eta_{ep}^1 \leq \eta_{ep}^2 < 1$ and $0 < \mu_1 < 1 < \mu_2$.

$L = L_k$
For $i = 1, ..., q$
    1. Selection of an iterate point
    Select $(x_{ep}^{i,k}, F(x_{ep}^{i,k}), \Delta_{ep}^{i,k}, \Delta_{sc}^{i,k}) \in L$ such that $x_{ep}^{i,k} \in \arg\min_{x \in L} f_i(x)$. Break ties by
    selecting the point with maximum extreme point trust-region radius corresponding to $f_i$.
    For all points $x^j$ in $L$, except the selected one, set $\Delta_{ep}^j(i) = 0$.
    If $\Delta_{ep}^{i,k}(i) < \Delta_{ep}^{min}$, stop the procedure and continue to the next $i$.

    2. Step calculation
    Compute the model function $m_i^k$, centered at $x_{ep}^{i,k}$, and $x_{ep}^{i,k*} \in \arg\min_{x \in B(x_{ep}^{i,k}, \Delta_{ep}^{i,k}(i))} m_i^k(x)$.
    If $m_i^k(x_{ep}^{i,k}) - m_i^k(x_{ep}^{i,k*}) = 0$ then set $\rho_{ep}^{i,k} = 0$. Else, set $\rho_{ep}^{i,k} = \frac{f_i(x_{ep}^{i,k}) - f_i(x_{ep}^{i,k*})}{m_i^k(x_{ep}^{i,k}) - m_i^k(x_{ep}^{i,k*})}$.

    3. Trial point acceptance and trust-region update
    If $\rho_{ep}^{i,k} \geq \eta_{ep}^1$ then:
        Set $\Delta_{ep}^{i,k*} = \Delta_{ep}^{i,k}$ and $\Delta_{sc}^{i,k*} = \Delta_{sc}^{i,k}$.
        If $\rho_{ep}^{i,k} \geq \eta_{ep}^2$ then set $\Delta_{ep}^{i,k*}(i) = \mu_2 * \Delta_{ep}^{i,k}(i)$.
        Set $\Delta_{ep}^{i,k}(i) = 0$.
        Add the new point to the list, by setting $L := L \cup \{(x_{ep}^{i,k*}, F(x_{ep}^{i,k*}), \Delta_{ep}^{i,k*}, \Delta_{sc}^{i,k*})\}$ and
        delete the dominated points from it.
    Else, set $\Delta_{ep}^{i,k}(i) := \mu_1 * \Delta_{ep}^{i,k}(i)$.
End For
$L_{k+1} = L$
```

At each iteration k, for each component of the objective function f_i, $i = 1, \dots, q$, the point $x_{ep}^{i,k}$, corresponding to the minimum value of f_i for the points in the list, is selected. Ties are broken by the largest extreme point trust-region radius corresponding to f_i, promoting successful iterations. Larger trust-region radii indicate points not yet selected or successful in their exploration. Once that $x_{ep}^{i,k}$ is selected, the extreme point trust-region radius corresponding to f_i is set equal to zero for all the other points in the list. They are no longer options as best candidates for the extreme point corresponding to the selected objective function component.

The quadratic Taylor model m_i^k (3), centered at $x_{ep}^{i,k}$, is computed and an iteration of a single objective trust-region algorithm is performed. The model is minimized in $B(x_{ep}^{i,k}, \Delta_{ep}^{i,k}(i))$, the closed ball centered at $x_{ep}^{i,k}$ with radius $\Delta_{ep}^{i,k}(i)$.

The ratio of agreement between the decrease obtained in the model and the variation obtained in the corresponding objective function component, $\rho_{ep}^{i,k}$, is computed, dictating the acceptance or rejection of the model minimizer, $x_{ep}^{i,k*}$, and the update strategy for the trust-region radius. Rules identical to the single objective case are adopted.

A high value of $\rho_{ep}^{i,k}$ indicates that the model is adequately predicting the reduction in the function value. The model minimizer will be accepted and the trust-region radius will be increased. In fact, when $\rho_{ep}^{i,k} > 0$, the value of the objective function $f_i(x_{ep}^{i,k*})$ is smaller than the one of $f_i(x_{ep}^{i,k})$. Thus, it is clear that $x_{ep}^{i,k*}$ is nondominated by all points in the list L_k. The point is added to the list, and all dominated points are removed from it. If $\rho_{ep}^{i,k}$ is low, then $x_{ep}^{i,k*}$ will not be accepted and the trust-region radius will be decreased, in an attempt of conferring more quality to the Taylor model as approximation to the real function component.

When $m_i^k(x_{ep}^{i,k}) - m_i^k(x_{ep}^{i,k*}) = 0$, preventing the computation of the agreement ratio, there was no model improvement, meaning that the current model center $x_{ep}^{i,k}$ is the model minimizer. At this situation, $\rho_{ep}^{i,k}$ is set equal to zero, forcing the decrease of the trust-region radius and the agreement between the Taylor model and the objective function component.

The described procedure for point acception/rejection and the update of the trust-region radius is identical to the single objective case. However, there is the possibility of $x_{ep}^{i,k}$ continue to be a nondominated point and still remain in the list. In this situation, it is no longer a candidate as extreme point of the objective function component f_i. Thus, $\Delta_{ep}^{i,k}(i)$ is set equal to zero.

2.2 Scalarization step

In the scalarization step, there is an attempt of filling the gaps in the current approximation to the Pareto front by selecting points associated with the largest gap for the objective function component under analysis. An adequate scalarization problem is solved and the corresponding minimizer is added to the list, if nondominated.

For each component of the objective function, $f_i, i = 1, \ldots, q$, a point $x_{sc}^{i,k}$, is selected, following the procedure described in Algorithm 3.

In the middle point step, there is an attempt of identifying the largest gap in the Pareto front, according to the objective function component under analysis, and filling it by considering the middle point of the line segment associated with the points defining the gap as the new iterate, $x_{sc}^{i,k}$. In case of ties between gaps, priority is given to the ones associated with the larger scalarization step trust-region radius, again promoting the progress of the algorithm. The procedure is repeated until a nondominated point has been found for the function component under analysis or all

Algorithm 3. Middle point step

```
Input
    Current list of nondominated points $L$, defined by (2).
    Current objective function component $i \in \{1, \dots, q\}$.
    The minimum trust-region radius value $\Delta_{sc}^{min}$.
    The initial trust-region radius values $\Delta_{ep}^{init}$ and $\Delta_{sc}^{init}$.

1. Compute and sort the gaps
Sort $\{f_i(x) \mid x \in L\}$ by increasing value. Compute the gaps between consecutive values
of the sorted $f_i$ and order them by decreasing value. Break ties according to the largest
scalarization step trust-region radius, corresponding to the points associated with the gaps.

For all gaps, starting from the largest one
    2. Compute the middle point
    If for at least one point of the pair associated with the gap, the corresponding
    trust-region radius associated to the scalarization step satisfies $\Delta_{sc} \geq \Delta_{sc}^{min}$ then:
      Compute $x_{middle}$, the middle point of the pair in the variable space.
    Else, continue to the next gap.

    3. Test the middle point
    If $x_{middle} \in L$, with the corresponding trust-region radius associated to the
    scalarization step satisfying $\Delta_{sc} \geq \Delta_{sc}^{min}$ then:
      Set $x_{sc}^{i,k} = x_{middle}$ and return.
    Else if $x_{middle} \notin L$ and is nondominated then:
      Set $x_{sc}^{i,k} = x_{middle}$, add it to the list by setting $L := L \cup \{(x_{sc}^{i,k}, F(x_{sc}^{i,k}), \Delta_{ep}^{init}, \Delta_{sc}^{init})\}$,
      delete the dominated points from it, and return.
    Else, continue to the next gap.
End For
```

the gaps have been exhausted. In the last case, the algorithm moves to the next objective function component. As it will be illustrated in the numerical results, reported in Sect. 4, this strategy is one of the key features of the new algorithm.

Once that $x_{sc}^{i,k}$ is selected, models are built for each objective function component, centered at the new point. A joint minimization of the models is performed, by solving the following scalarization problem, computing the new point $x_{sc}^{i,k*}$:

$$\begin{array}{ll} \min & t \\ \text{s.t.} & m_l^k(x) - m_l^k(x_{sc}^{i,k}) \leq t, \; l = 1, \dots, q, \\ & x \in B(x_{sc}^{i,k}, \Delta_{sc}^{i,k}), \\ & t \in \mathbb{R}. \end{array} \tag{4}$$

A similar scalarization approach was considered in [26] and [4], in the last case enriched with an extra set of linear inequality constraints. Positive definite Hessians are required by [26], for the theoretical analysis of the algorithm. In both cases, the proposed algorithms generate a single point, with no explicit attempt of computing any approximation to the complete Pareto front of the problem.

The criterion for accepting $x_{sc}^{i,k*}$ and the update rules for the scalarization step trust-region radius are the ones of [30, 31]. The auxiliary functions

$\phi(x) = \max_{j=1,\ldots,q} f_j(x)$ and $\phi_m^k(x) = \max_{j=1,\ldots,q} m_j^k(x)$ are defined and used to compute the ratio

$$\rho_{sc}^{i,k} = \frac{\phi(x_{sc}^{i,k}) - \phi(x_{sc}^{i,k*})}{\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*})}.$$

Algorithm 4. Scalarization step

```
Input
        Current list of nondominated points $L_k$, defined by (2).
        The minimum trust-region radius value $\Delta_{sc}^{min}$.
        Values for the parameters $0 < \eta_{sc}^1 \leq \eta_{sc}^2 < 1$ and $0 < \mu_1 < 1 < \mu_2$.

$L = L_k$
For $i = 1, ..., q$
        1. Selection of an iterate point
        Compute $p = |\{x \in L \, : \, \Delta_{sc} \geq \Delta_{sc}^{min}\}|$.
        If $p = 0$ then stop the procedure and continue to the next $i$.
        If $p = 1$ then set $x_{sc}^{i,k}$ equal to the point in $L$ satisfying $\Delta_{sc} \geq \Delta_{sc}^{min}$.
        If $p \geq 2$ then go to the Middle Point Step to compute $x_{sc}^{i,k}$.

        2. Step calculation
        Compute the model functions $m_l^k, l = 1, \ldots, q$, centered at $x_{sc}^{i,k}$, and $x_{sc}^{i,k*}$ by
        solving the scalarization Problem (4).
        Define $\phi(x) = \max_{j=1,\ldots,q} f_j(x)$ and $\phi_m^k(x) = \max_{j=1,\ldots,q} m_j^k(x)$.
        If $\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) = 0$ then set $\rho_{sc}^{i,k} = 0$. Else compute
        $\rho_{sc}^{i,k} = \frac{\phi(x_{sc}^{i,k}) - \phi(x_{sc}^{i,k*})}{\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*})}$.

        3. Trial point acceptance and trust-region update
        If $\rho_{sc}^{i,k} \geq \eta_{sc}^1$ and $x_{sc}^{i,k*}$ is nondominated then:
          If $\rho_{sc}^{i,k} \geq \eta_{sc}^2$ then set $\Delta_{sc}^{i,k} := \mu_2 * \Delta_{sc}^{i,k}$.
          Set $\Delta_{ep}^{i,k*} = \Delta_{ep}^{i,k}$ and $\Delta_{sc}^{i,k*} = \Delta_{sc}^{i,k}$.
          Add the new point to the list, by setting
        $L := L \cup \{(x_{sc}^{i,k*}, F(x_{sc}^{i,k*}), \Delta_{ep}^{i,k*}, \Delta_{sc}^{i,k*})\}$ and
          delete the dominated points from it.
        Else, set $\Delta_{sc}^{i,k} := \mu_1 * \Delta_{sc}^{i,k}$.
End For
$L_{k+1} = L$
```

In [30], it is established that when $\rho_{sc}^{i,k} > 0$, descent is guaranteed for at least one component of the objective function. When the denominator $\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*})$ is equal to zero, $\rho_{sc}^{i,k}$ is set equal to zero, forcing the decrease of the scalarization step trust-region radius and consequently the agreement between the models and the objective function components.

Having defined the ratio $\rho_{sc}^{i,k}$, the strategy to accept/reject the new point and to update the trust-region radius is similar to the one of any trust-region method, with the exception that successful steps are obligatory associated to nondominated points. If $\rho_{sc}^{i,k}$ is large enough but the new point is dominated, the trust-region radius is reduced to increase the model quality. Algorithm 4 details the procedure.

3 Convergence analysis

For analyzing the theoretical behavior of the algorithm, we will consider that no stoping criteria are defined. In particular, $\Delta_{ep}^{min} = \Delta_{sc}^{min} = 0$. Convergence will be established for linked sequences of points $\{x^k\}_{k\in K}$ generated by MOTR, a concept introduced in [19], in a Derivative-free Optimization setting.

Definition 3.1 Consider $\{L_k\}_{k\in\mathbb{N}}$ the sequence of sets of nondominated points generated by Algorithm 1. A linked sequence is a sequence $\{x^k\}_{k\in K}$, where $K \subseteq \mathbb{N}$ denotes the indexes of the points belonging to the linked sequence, such that for any $k \in K$, the element $(x^k, F(x^k), \Delta_{ep}^k, \Delta_{sc}^k) \in L_k$ is generated from the element $(x^{k-1}, F(x^{k-1}), \Delta_{ep}^{k-1}, \Delta_{sc}^{k-1}) \in L_{k-1}$.

An initial point of a linked sequence can be:

- A point in the initial list, provided by the user;
- Middle points generated by MOTR in the middle point step, such that these points are not currently in the list and are added to the list by the algorithm.

We are going to establish that every linked sequence generated by MOTR converges to a Pareto critical point, a necessary condition for being a solution of Problem (1), formalized in the following definition.

Definition 3.2 Let $f_i : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable functions, for $i \in \{1, \ldots, q\}$. A point $x^* \in \mathbb{R}^n$ is said to be Pareto critical for Problem (1) if

$$\forall d \in \mathbb{R}^n,\ \exists i \in \{1, \ldots, q\} : \nabla f_i(x^*)^\top d \geq 0.$$

The next lemma, originally stated in [14], provides a criticality measure for multiobjective optimization.

Lemma 3.3 *Let $f_i : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable functions, for $i \in \{1, \ldots, q\}$. Define*

$$\omega(x) = - \min_{\|d\|\leq 1} \max_{i=1,\ldots,q} \nabla f_i(x)^\top d. \tag{5}$$

The following statements hold:

- *The mapping $x \mapsto \omega(x)$ is continuous;*
- *$\omega(x) \geq 0$ for all $x \in \mathbb{R}^n$;*
- *A point $x^* \in \mathbb{R}^n$ is Pareto critical if and only if $\omega(x^*) = 0$.*

Under reasonable assumptions, we are going to establish that for every linked sequence $\{x^k\}_{k\in K}$ generated by MOTR,

$$\lim_{k\to+\infty;\, k\in K} \omega(x^k) = 0.$$

The structure of the proof follows the arguments considered by [4, 30, 31].

For $i \in \{1,\ldots,q\}$, we assume that the objective function component f_i is lower bounded and twice continuously differentiable, with uniformly bounded Hessian. Thus, function $\phi(x) = \max_{i=1,\ldots,q} f_i(x)$ is also lower bounded.

Assumption 3.1 For $i \in \{1,\ldots,q\}$, the Hessian matrix of the objective function component f_i is uniformly bounded, meaning that there is a constant $\kappa_h > 0$ such that

$$\|\nabla^2 f_i(x)\| \le \kappa_h$$

for all $x \in \mathbb{R}^n$ and for all $i \in \{1,\ldots,q\}$.

Assumption 3.1 implies that ∇f_i is Lipschitz continuous, for each $i \in \{1,\ldots,q\}$. From it, we can deduce that function ω, defined by (5), is uniformly continuous [31].

At each iteration k, for $i \in \{1,\ldots,q\}$, model m_i^k, centered at x^k, is a quadratic Taylor model, again twice continuously differentiable and satisfying:

$$\begin{aligned} m_i^k(x^k) &= f_i(x^k),\\ \nabla m_i^k(x^k) &= \nabla f_i(x^k),\\ \nabla^2 m_i^k(x^k) &= \nabla^2 f_i(x^k). \end{aligned}$$

In this paper, B_k denotes the current iteration ball, defined as

$$B_k = B(x^k, \Delta_k) = \left\{x \in \mathbb{R}^n \mid \|x - x^k\| \le \Delta_k\right\},$$

where x^k is the current iterate and Δ_k represents the current trust-region radius.

Assumption 3.1 allows us to establish the well-known error bounds for Taylor models.

Lemma 3.4 *Let Assumption 3.1 hold. At every iteration k, the model m_i^k is valid for f_i in B_k, for all $i \in \{1,\ldots,q\}$, that is, there exists a constant $\kappa_{fm} > 0$ such that*

$$|f_i(x) - m_i^k(x)| \le \kappa_{fm}\Delta_k^2$$

holds for all $x \in B_k$.

Function ω, defined by equation (5), can be generalized to models through:

$$\omega_{m^k}(x) = -\min_{\|d\|\le 1}\max_{i=1,\ldots,q} \nabla m_i^k(x)^\top d.$$

The use of Taylor models guarantees that $\omega_{m^k}(x^k) = \omega(x^k)$, at each iteration k, where x^k represents the point where the model was built. This equality ensures that when the iteration point x^k is Pareto critical or close to criticality for the model, the same applies to the objective function.

To prove convergence, model minimization should provide a sufficient decrease at each iteration. Following [31], for the scalarization step, we quantify the best model reduction obtained along a direction belonging to $\mathcal{D}(x)$, the set of directions associated with the solution of (5), within the trust-region B_k. For $i \in \{1, \dots, q\}$, let $d_k^* \in \mathcal{D}(x_{sc}^{i,k})$ and compute α_k by solving

$$\min_{\alpha \geq 0} \{\phi_m^k(x_{sc}^{i,k} + \alpha d_k^*) : x_{sc}^{i,k} + \alpha d_k^* \in B_k\}. \tag{6}$$

The Pareto-Cauchy point is defined as

$$x_k^C = x_{sc}^{i,k} + d_k^C, \tag{7}$$

where $d_k^C := \alpha_k d_k^*$ and $B_k = B(x_{sc}^{i,k}, \Delta_{sc}^{i,k})$.

Lemma 3.5 *Let Assumption 3.1 hold. For $i \in \{1, \dots, q\}$ and $k \in \mathbb{N}$, the Pareto-Cauchy point $x_k^C = x_{sc}^{i,k} + d_k^C$, defined by (7), satisfies*

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_k^C) \geq \frac{1}{2}\omega(x_{sc}^{i,k}) \min\left\{\frac{\omega(x_{sc}^{i,k})}{\kappa_h}, \Delta_{sc}^{i,k}\right\}. \tag{8}$$

Proof Since $d_k^* \in \mathcal{D}(x_{sc}^{i,k})$, we have $\|d_k^*\| \leq 1$, which implies that, for all $\alpha \in [0, \Delta_{sc}^{i,k}]$, $x_{sc}^{i,k} + \alpha d_k^* \in B_k$. Problem (6) has the same solution than

$$\max_{0 \leq \alpha \leq \Delta_{sc}^{i,k}} \{\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k} + \alpha d_k^*)\}.$$

On the other hand, for all $\alpha \geq 0$, we have

$$\begin{aligned} \phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k} + \alpha d_k^*) &= \max_{j=1,\dots,q} m_j^k(x_{sc}^{i,k}) - \max_{j=1,\dots,q} m_j^k(x_{sc}^{i,k} + \alpha d_k^*) \\ &\geq - \max_{j=1,\dots,q} \alpha \nabla {m_j^k}^\top(x_{sc}^{i,k}) d_k^* - \max_{j=1,\dots,q} \frac{1}{2}\alpha^2 {d_k^*}^\top \nabla^2 m_j^k(x_{sc}^{i,k}) d_k^*. \end{aligned}$$

According to (5), Assumption 3.1, $\|d_k^*\| \leq 1$, and the Cauchy–Schwarz inequality we have

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k} + \alpha d_k^*) \geq \alpha\omega(x_{sc}^{i,k}) - \frac{1}{2}\alpha^2\kappa_h.$$

Then, it is clear that

$$\max_{0 \leq \alpha \leq \Delta_{sc}^{i,k}} \left\{\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k} + \alpha d_k^*)\right\} \geq \max_{0 \leq \alpha \leq \Delta_{sc}^{i,k}} \left\{\alpha\omega(x_{sc}^{i,k}) - \frac{1}{2}\alpha^2\kappa_h\right\}.$$

In other words,

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k} + d_k^C) \geq \max_{0 \leq \alpha \leq \Delta_{sc}^{i,k}} \left\{ \alpha\omega(x_{sc}^{i,k}) - \frac{1}{2}\alpha^2\kappa_h \right\}.$$

Let us consider the concave function g, defined by $g(\alpha) = \alpha\omega(x_{sc}^{i,k}) - \frac{1}{2}\alpha^2\kappa_h$, with unconstrained maximizer $\alpha^* = \frac{\omega(x_{sc}^{i,k})}{\kappa_h} \geq 0$, corresponding to the optimum value $g(\alpha^*) = \frac{1}{2}\frac{\omega(x_{sc}^{i,k})^2}{\kappa_h} \geq 0$. Two cases can occur:

- If $0 \leq \alpha^* \leq \Delta_{sc}^{i,k}$ then $\max_{0 \leq \alpha \leq \Delta_{sc}^{i,k}} \left\{ \alpha\omega(x_{sc}^{i,k}) - \frac{1}{2}\alpha^2\kappa_h \right\} = \frac{1}{2}\frac{\omega(x_{sc}^{i,k})^2}{\kappa_h}$;
- If $\alpha^* > \Delta_{sc}^{i,k}$ then $\max_{0 \leq \alpha \leq \Delta_{sc}^{i,k}} \left\{ \alpha\omega(x_{sc}^{i,k}) - \frac{1}{2}\alpha^2\kappa_h \right\} = \Delta_{sc}^{i,k}\omega(x_{sc}^{i,k}) - \frac{1}{2}(\Delta_{sc}^{i,k})^2\kappa_h$.

In this last case, since $\alpha^* = \frac{\omega(x_{sc}^{i,k})}{\kappa_h} > \Delta_{sc}^{i,k}$, we have $\Delta_{sc}^{i,k}\omega(x_{sc}^{i,k}) - \frac{1}{2}(\Delta_{sc}^{i,k})^2\kappa_h \geq \frac{1}{2}\Delta_{sc}^{i,k}\omega(x_{sc}^{i,k})$, resulting in

$$\begin{aligned}\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_k^C) &\geq \min\left\{ \frac{1}{2}\frac{\omega(x_{sc}^{i,k})^2}{\kappa_h}, \frac{1}{2}\Delta_{sc}^{i,k}\omega(x_{sc}^{i,k}) \right\} \\ &= \frac{1}{2}\omega(x_{sc}^{i,k}) \min\left\{ \frac{\omega(x_{sc}^{i,k})}{\kappa_h}, \Delta_{sc}^{i,k} \right\}.\end{aligned}$$

□

Let $(x_{sc}^{i,k*}, t^*)$ be the solution of Problem (4). The following lemma states an important property of t^*.

Lemma 3.6 *At each iteration k and for each i* $\in \{1, \ldots, q\}$, $x_{sc}^{i,k}$ *is not a Pareto critical point for* $\min_{x \in B_k}(m_1^k(x), \ldots, m_q^k(x))$, *if and only if* $t^* < 0$.

Proof Since $(x_{sc}^{i,k}, 0)$ is feasible for Problem (4), $t^* \leq 0$. It is clear that $x_{sc}^{i,k}$ is not a Pareto critical point when $t^* < 0$.

Now, assume that $x_{sc}^{i,k}$ is not a Pareto critical point. Then, it is not a weakly efficient point, meaning that there exists a point $x' \in B_k$ such that for all $j \in \{1, \ldots, q\}$, $m_j^k(x') < m_j^k(x_{sc}^{i,k})$. So,

$$m_j^k(x^{'}) - m_j^k(x_{sc}^{i,k}) < 0, \quad \forall j \in \{1, \dots, q\}.$$

Considering $t^{'} = \max_{j=1,\dots,q}(m_j^k(x^{'}) - m_j^k(x_{sc}^{i,k}))$, we have

$$m_j^k(x^{'}) - m_j^k(x_{sc}^{i,k}) \leq t^{'} < 0, \quad \forall j \in \{1, \dots, q\}.$$

Hence, t^* should be strictly negative because $(x^{'}, t^{'})$ is feasible for Problem (4). □

Lemma 3.7 *Let Assumption 3.1 hold. At each iteration k and for each i* $\in \{1, \dots, q\}$, *there exists* $j \in \mathbb{N}$ *such that*

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq \left(\frac{1}{2}\right)^j \omega(x_{sc}^{i,k}) \min\left\{\frac{\omega(x_{sc}^{i,k})}{\kappa_h}, \Delta_{sc}^{i,k}\right\},$$

where $(x_{sc}^{i,k*}, t^*)$ *is the solution of Problem* (4).

Proof Two different cases need to be analyzed. Assume that $x_{sc}^{i,k}$ is not Pareto critical. According to Lemma 3.6, t^* is strictly negative. So, for each $l \in \{1, \dots, q\}$, we have

$$m_l^k(x_{sc}^{i,k}) - m_l^k(x_{sc}^{i,k*}) \geq -t^* > 0.$$

By considering $\phi_m^k(x) = \max_{l=1,\dots,q} m_l^k(x)$, it results

$$\phi_m^k(x_{sc}^{i,k}) - m_l^k(x_{sc}^{i,k*}) \geq m_l^k(x_{sc}^{i,k}) - m_l^k(x_{sc}^{i,k*}), \text{ for all } l \in \{1, \dots, q\}.$$

Let j be the index such that $\phi_m^k(x_{sc}^{i,k*}) = m_j^k(x_{sc}^{i,k*})$. Then

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq m_j^k(x_{sc}^{i,k}) - m_j^k(x_{sc}^{i,k*}).$$

Hence,

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq -t^* > 0. \tag{9}$$

Since $\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_k^C) \geq 0$, there must exist $j \in \mathbb{N}$ such that

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq \left(\frac{1}{2}\right)^{j-1} (\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_k^C)).$$

Hence, considering (8), it implies

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq \left(\frac{1}{2}\right)^j \omega(x_{sc}^{i,k}) \min\left\{\frac{\omega(x_{sc}^{i,k})}{\kappa_h}, \Delta_{sc}^{i,k}\right\}.$$

If $x_{sc}^{i,k}$ is Pareto critical, then $\omega(x_{sc}^{i,k}) = 0$. So, the right side of this inequality is equal to zero and, since $\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq 0$, the inequality holds. □

The last three lemmas motivate us to consider the following assumption, stating that, at each scalarization step, a sufficient reduction in the model is ensured.

Assumption 3.2 There is a constant $\kappa_\phi \in (0,1)$ such that at each iteration k, where the scalarization step is performed, for all $i \in \{1,\ldots,q\}$, we have

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq \kappa_\phi \, \omega(x_{sc}^{i,k}) \min\left\{\frac{\omega(x_{sc}^{i,k})}{\kappa_h}, \Delta_{sc}^{i,k}\right\}. \tag{10}$$

As long as $x_{sc}^{i,k}$ is not Pareto critical, the left side of (10) is strictly positive.

The following lemma provides an error bound for ϕ_m^k as approximation of ϕ and is a keystone to establish convergence.

Lemma 3.8 *Let Assumption 3.1 hold. At every iteration k, the model ϕ_m^k is valid for ϕ at $x_{sc}^{i,k*}$, for all $i \in \{1,\ldots,q\}$, that is*

$$\left|\phi(x_{sc}^{i,k*}) - \phi_m^k(x_{sc}^{i,k*})\right| \leq \kappa_{fm}(\Delta_{sc}^{i,k})^2,$$

where κ_{fm} is defined in Lemma 3.4.

Proof Two situations should be analyzed. Assume that $\phi(x_{sc}^{i,k*}) \geq \phi_m^k(x_{sc}^{i,k*})$. Consider $j \in \{1,\ldots,q\}$ such that $\phi(x_{sc}^{i,k*}) = f_j(x_{sc}^{i,k*})$. Using Lemma 3.4 and the fact that $\phi_m^k(x_{sc}^{i,k*}) \geq m_j^k(x_{sc}^{i,k*})$, we have

$$\begin{aligned}\left|\phi(x_{sc}^{i,k*}) - \phi_m^k(x_{sc}^{i,k*})\right| &= f_j(x_{sc}^{i,k*}) - \phi_m^k(x_{sc}^{i,k*}) \leq f_j(x_{sc}^{i,k*}) - m_j^k(x_{sc}^{i,k*}) \\ &\leq \kappa_{fm}(\Delta_{sc}^{i,k})^2.\end{aligned}$$

Assume now that $\phi(x_{sc}^{i,k*}) < \phi_m^k(x_{sc}^{i,k*})$. Consider $j \in \{1,\ldots,q\}$ such that $\phi_m^k(x_{sc}^{i,k*}) = m_j^k(x_{sc}^{i,k*})$. Again, Lemma 3.4 and the fact that $\phi(x_{sc}^{i,k*}) \geq f_j(x_{sc}^{i,k*})$ allow us to conclude that

$$\begin{aligned}\left|\phi(x_{sc}^{i,k*}) - \phi_m^k(x_{sc}^{i,k*})\right| &= m_j^k(x_{sc}^{i,k*}) - \phi(x_{sc}^{i,k*}) \\ &\leq m_j^k(x_{sc}^{i,k*}) - f_j(x_{sc}^{i,k*}) \leq \kappa_{fm}(\Delta_{sc}^{i,k})^2.\end{aligned}$$

□

In the remaining of the analysis, we classify the successful scalarization step iterations. The set of indexes of *successful scalarization step iterations* is denoted by

$$S = \left\{(k,i),\ k \in \mathbb{N}, i \in \{1,\ldots,q\} : k \text{ is a scalarization step iteration and } \rho_{sc}^{i,k} \geq \eta_{sc}^1\right\}$$

and the set of indexes of *very successful scalarization step iterations* corresponds to

$$V = \left\{(k,i),\ k \in \mathbb{N}, i \in \{1,\ldots,q\} : k \text{ is a scalarization step iteration and } \rho_{sc}^{i,k} \geq \eta_{sc}^2\right\}.$$

Two different scenarios may occur for each linked sequence of points $\{x^k\}_{k\in K}$ generated by MOTR:

1. There exists $i \in \{1, \dots, q\}$, such that for each $k \in \mathbb{N}$,

$$\Delta_{ep}^{i,k} > 0.$$

2. For each $i \in \{1, \dots, q\}$, there exists $k_i \in \mathbb{N}$, such that for all $k > k_i$,

$$\Delta_{ep}^{i,k} = 0.$$

Remark 3.1 In the first scenario, the linked sequence, updated at the extreme point step, matches the set of iterates generated by a single objective trust-region method, when applied to the objective function component f_i. Stationarity is then guaranteed for f_i and the corresponding limit point is a Pareto critical point. The proof is similar to the single objective trust-region case (see [5, 18, 24]).

Remark 3.2 In the second scenario, define $k_{ep} = \max\{k_i \mid i = 1, \dots, q\}$. For $k > k_{ep}$, it holds $\Delta_{ep}^{i,k} = 0,\ \forall i = 1, \dots, q$. Therefore, for $k > k_{ep}$, all points of the linked sequence have been generated in the scalarization step. The remaining of the analysis focuses on this situation.

The following two lemmas clarify the behavior of MOTR when the current point is not Pareto critical.

Lemma 3.9 *Let Assumptions 3.1 and 3.2 hold. Suppose that at the scalarization step iteration k, for i* $\in \{1, \dots, q\}$, $x_{sc}^{i,k}$ *is not a Pareto critical point and*

$$\Delta_{sc}^{i,k} \leq \frac{\kappa_\phi \omega(x_{sc}^{i,k})(1-\eta_{sc}^2)}{\kappa_v}, \tag{11}$$

with $\kappa_v = \max\{\kappa_{fm}, \kappa_h\}$. *Then the pair* (k, i) *corresponds to a very successful scalarization step iteration, and* $\Delta_{sc}^{i,k*} > \Delta_{sc}^{i,k}$.

Proof According to Lemma 3.3, $\omega(x_{sc}^{i,k}) > 0$, because $x_{sc}^{i,k}$ is not a Pareto critical point. On the other hand, $\kappa_\phi, \eta_{sc}^2 \in (0, 1)$. Thus, $\kappa_\phi(1-\eta_{sc}^2) \in (0, 1)$ and

$$\Delta_{sc}^{i,k} \leq \frac{\kappa_\phi \omega(x_{sc}^{i,k})(1-\eta_{sc}^2)}{\kappa_v} < \frac{\omega(x_{sc}^{i,k})}{\kappa_v}.$$

From Assumption 3.2, we have

$$\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*}) \geq \kappa_\phi \omega(x_{sc}^{i,k}) \min\left\{\frac{\omega(x_{sc}^{i,k})}{\kappa_h}, \Delta_{sc}^{i,k}\right\} = \kappa_\phi \omega(x_{sc}^{i,k})\Delta_{sc}^{i,k}.$$

Considering this inequality, the equation $\rho_{sc}^{i,k} = \frac{\phi(x_{sc}^{i,k})-\phi(x_{sc}^{i,k*})}{\phi_m^k(x_{sc}^{i,k})-\phi_m^k(x_{sc}^{i,k*})}$, Lemma 3.8, and (11), we have

$$|\rho_{sc}^{i,k} - 1| = \left|\frac{\phi_m^k(x_{sc}^{i,k*}) - \phi(x_{sc}^{i,k*})}{\phi_m^k(x_{sc}^{i,k}) - \phi_m^k(x_{sc}^{i,k*})}\right| \leq \frac{\kappa_{fm}}{\kappa_\phi \omega(x_{sc}^{i,k})}\Delta_{sc}^{i,k} \leq \frac{\kappa_v}{\kappa_\phi \omega(x_{sc}^{i,k})}\Delta_{sc}^{i,k} \leq (1-\eta_{sc}^2).$$

Consequently, $\rho_{sc}^{i,k} \geq \eta_{sc}^2$. So, the pair (k, i) corresponds to a very successful scalarization step iteration and $\Delta_{sc}^{i,k*} > \Delta_{sc}^{i,k}$. □

The following lemma states that, as long as $x_{sc}^{i,k}$ is not Pareto critical, the scalarization step trust-region radius can not be too small. In fact, it should be lower bounded by a strictly positive constant.

Lemma 3.10 *Let Assumptions 3.1 and 3.2 hold and consider the constant* $\sigma > 0$. *If* $\omega(x_{sc}^{i,k}) \geq \sigma$ *holds for the pair* (k, i), *with* k *a scalarization step iteration and* $i \in \{1, \ldots, q\}$, *then there is a constant* $\Delta > 0$, *depending on* σ, *such that* $\Delta_{sc}^{i,k} \geq \Delta$.

Proof Assume, as mean of a contradiction, that for each $\Delta > 0$ there is a pair (k, i), with k a scalarization step iteration and $i \in \{1, \ldots, q\}$, satisfying $\omega(x_{sc}^{i,k}) \geq \sigma > 0$, such that

$$\Delta_{sc}^{i,k} < \Delta.$$

In particular, consider

$$\Delta = \frac{\mu_1 \sigma \kappa_\phi (1-\eta_{sc}^2)}{\kappa_v},$$

with $\kappa_v = \max\{\kappa_{fm}, \kappa_h\}$. Let $(\bar{k}, i)$ be the first pair such that $\omega(x_{sc}^{i,\bar{k}}) \geq \sigma > 0$ and

$$\Delta_{sc}^{i,\bar{k}^*} < \frac{\mu_1 \sigma \kappa_\phi (1-\eta_{sc}^2)}{\kappa_v}.$$

Then, it holds $\Delta_{sc}^{i,\bar{k}^*} < \Delta_{sc}^{i,\bar{k}}$. Thus,

$$\Delta_{sc}^{i,\bar{k}} = \frac{\Delta_{sc}^{i,\bar{k}^*}}{\mu_1} < \frac{\sigma \kappa_\phi (1-\eta_{sc}^2)}{\kappa_v} \leq \frac{\omega(x_{sc}^{i,\bar{k}})\kappa_\phi (1-\eta_{sc}^2)}{\kappa_v}.$$

Since point $x_{sc}^{i,\bar{k}}$ is not Pareto critical, according to Lemma 3.9, the pair $(\bar{k}, i)$ corresponds to a very successful scalarization step iteration, and $\Delta_{sc}^{i,\bar{k}^*} > \Delta_{sc}^{i,\bar{k}}$. This contradicts $\Delta_{sc}^{i,\bar{k}^*} < \Delta_{sc}^{i,\bar{k}}$ and the initial assumption. □

Considering Remarks 3.1 and 3.2, the following lemma states the first convergence result for linked sequences of MOTR, generated at the scalarization step.

Lemma 3.11 *Suppose that Assumptions 3.1 and 3.2 hold. Let $\{x^k\}_{k\in K}$ be a linked sequence generated by MOTR at the scalarization step, with finitely many successful iterations at the scalarization step. Then this linked sequence converges to a Pareto critical point.*

Proof Assume that $x^{k_0+l} = x^*$ for all $l \in \mathbb{N}$, where $k_0 = \max\{k_s, k_{ep}\}$, k_s is the index of the last successful scalarization step iteration and k_{ep} is defined as in Remark 3.2. So, $\Delta_{sc}^{k_0+l}$ converges to zero, because in the scalarization step all iterations are unsuccessful, for sufficiently large l.

Suppose that x^* is not a Pareto critical point. According to Lemma 3.9, there must be a very successful scalarization step iteration, with index larger than k_0, which is a contradiction because all iterations after k_0 are unsuccessful. Therefore, x^* is a Pareto critical point. □

The following lemma clarifies the behavior of MOTR, when a linked sequence has an infinite number of distinct points generated at the scalarization step.

Lemma 3.12 *Suppose that Assumptions 3.1 and 3.2 hold. Let $\{x^k\}_{k\in K}$ be a linked sequence of points generated by MOTR, with infinitely many successful iterations at the scalarization step. Then*

$$\liminf_{k\to+\infty;\, k\in K} \omega(x^k) = 0.$$

Proof Suppose that $\liminf_{k\to+\infty\,;\, k\in K} \omega(x^k) \neq 0$. So, there must exist a constant $\epsilon > 0$ such that for all $k \in K$,

$$\omega(x^k) \geq \epsilon.$$

Lemma 3.10 guarantees the existence of $\Delta > 0$ such that $\Delta_{sc}^k \geq \Delta$, for all $k \in K$.

Consider S, the set of indexes of successful scalarization step iterations, $k \in S \cap K$, and $k > k_{eps}$, where k_{eps} is the index of the first successful scalarization step iteration after k_{ep}, and k_{ep} is defined as in Remark 3.2. Thus, $\rho_{sc}^k \geq \eta_{sc}^1$. According to Assumption 3.2, we have

$$\begin{aligned}\phi(x^k) - \phi(x^{k+1}) &\geq \eta_{sc}^1(\phi_m^k(x^k) - \phi_m^k(x^{k+1}))\\ &\geq \eta_{sc}^1\kappa_\phi\omega(x^k)\min\left\{\frac{\omega(x^k)}{\kappa_h}, \Delta_{sc}^k\right\} \geq \eta_{sc}^1\kappa_\phi\epsilon\min\left\{\frac{\epsilon}{\kappa_h}, \Delta\right\}.\end{aligned}$$

Summing over all successful iterations at the scalarization step, from k_{eps} to k results in

$$\begin{aligned}\phi(x^{k_{eps}}) - \phi(x^{k+1}) &= \sum_{i=k_{eps}, i\in S\cap K}^{k} \phi(x^i) - \phi(x^{i+1})\\ &\geq \sigma_k\, \eta_{sc}^1\, \kappa_\phi\, \epsilon\, \min\left\{\frac{\epsilon}{\kappa_h}, \Delta\right\},\end{aligned}$$

where σ_k represents the number of successful scalarization step iterations in the linked sequence from k_{eps} to k. It is clear that $\lim\limits_{k\to+\infty\,;\,k\in K} \sigma_k = +\infty$, because there are infinitely many such iterations. Thus, $\phi(x^{k_{ep}}) - \phi(x^{k+1})$ is unbounded. So, $\phi(x)$ can not be bounded from below and this is a contradiction. Therefore, $\liminf\limits_{k\to+\infty\,;\,k\in K} \omega(x^k) = 0$. □

We are now ready to prove the main result, for linked sequences generated by MOTR.

Theorem 3.1 *Let Assumptions 3.1 and 3.2 hold. For every linked sequence of points $\{x^k\}_{k\in K}$ generated by MOTR, we have*

$$\lim_{k\to+\infty;\,k\in K} \omega(x^k) = 0.$$

Proof Let $\{x^k\}_{k\in K}$ be a linked sequence generated by MOTR. If there exists $i \in \{1,\dots,q\}$, such that for each $k \in \mathbb{N}$, $\Delta_{ep}^{i,k} > 0$ or if for each $i \in \{1,\dots,q\}$, there exists $k_i \in \mathbb{N}$, such that for all $k > k_i$, $\Delta_{ep}^{i,k} = 0$, but there are only finitely many successful iterations at the scalarization step, Remarks 3.1, 3.2, and Lemma 3.11, guarantee the convergence of $\{x^k\}_{k\in K}$ to a Pareto critical point, which implies $\lim\limits_{k\to+\infty\,;\,k\in K} \omega(x^k) = 0$, by Lemma 3.3.

Now, assume that for all $k > k_i$, $\Delta_{ep}^{i,k} = 0$ and there is an infinite number of successful scalarization step iterations. Suppose that there exists a subsequence of successful scalarization step iterations, indexed by $\{t_j > k_{ep}\} \subset S \cap K$, such that

$$\omega(x^{t_j}) \geq 2\epsilon > 0, \tag{12}$$

for some $\epsilon > 0$ and for all j, where k_{ep} is defined as in Remark 3.2.

From Lemma 3.12, for each t_j, there exists a first successful scalarization step iteration $l_j > t_j$ such that $\omega(x^{l_j}) < \epsilon$. Thus, there exists another subsequence of $S \cap K$, indexed by $\{l_j\}$, such that

$$\omega(x^k) \geq \epsilon \text{ for } t_j \leq k < l_j \text{ and } \omega(x^{l_j}) < \epsilon. \tag{13}$$

Consider the successful iterates whose indexes are in

$$\mathcal{K} = \{k \in S \cap K \mid \exists j \in \mathbb{N} : t_j \leq k < l_j\},$$

where t_j and l_j belong to the two subsequences defined above.

Assumption 3.2, the fact that $\mathcal{K} \subset S \cap K$, and inequalities (13) guarantee that, for $k \in \mathcal{K}$,

$$\begin{aligned} \phi(x^k) - \phi(x^{k+1}) &\geq \eta_{sc}^1(\phi_m^k(x^k) - \phi_m^k(x^{k+1})) \\ &\geq \eta_{sc}^1 \kappa_\phi \omega(x^k) \min\left\{\frac{\omega(x^k)}{\kappa_h}, \Delta_{sc}^k\right\} \\ &\geq \eta_{sc}^1 \kappa_\phi \epsilon \min\left\{\frac{\epsilon}{\kappa_h}, \Delta_{sc}^k\right\}. \end{aligned} \tag{14}$$

The sequence $\{\phi(x^k)\}_{k\in K}$ is convergent, since it is monotonically decreasing and bounded from below. Thereby, $\lim\limits_{k\to+\infty\,;\,k\in K} \left(\phi(x^k) - \phi(x^{k+1})\right) = 0$. Consequently, considering the minimum part in the last term of (14), for $k \in \mathcal{K}$ sufficiently large, it implies that

$$\Delta_{sc}^k \leq \frac{1}{\eta_{sc}^1 \kappa_\phi \epsilon}(\phi(x^k) - \phi(x^{k+1})).$$

Therefore, for j sufficiently large, it holds that

$$\begin{aligned} \|x^{t_j} - x^{l_j}\| &\leq \sum_{i=t_j, i\in\mathcal{K}}^{l_j-1} \|x^i - x^{i+1}\| \\ &\leq \sum_{i=t_j, i\in\mathcal{K}}^{l_j-1} \Delta_{sc}^i \\ &\leq \frac{1}{\eta_{sc}^1 \kappa_\phi \epsilon}(\phi(x^{t_j}) - \phi(x^{l_j})). \end{aligned}$$

Again, the convergence of $\{\phi(x^k)\}_{k\in K}$ implies

$$\lim_{j\to+\infty} \|x^{t_j} - x^{l_j}\| = 0.$$

Assumption 3.1 and the uniform continuity of ω allow us to conclude that

$$\lim_{j\to+\infty} |\omega(x^{t_j}) - \omega(x^{l_j})| = 0,$$

contradicting the fact that $|\omega(x^{t_j}) - \omega(x^{l_j})| \geq \epsilon$, a consequence of the definition of sequences $\{t_j\}$ and $\{l_j\}$, in (12) and (13).

So, no subsequence of successful iterations satisfying (12) can exist and $\lim\limits_{k\to+\infty\,;\,k\in K} \omega(x^k) = 0$. □

4 Numerical results

The numerical experiments were conducted with two main goals. The first was to illustrate the importance of each of the key algorithmic features of MOTR, namely the extreme point step, the scalarization step, and the middle point strategy, corresponding to Algorithms 2, 4, and 3, respectively. Even if not required for establishing convergence, the middle point strategy is relevant for the numerical performance of the algorithm.

With this purpose, three different versions of MOTR were implemented, each omitting one of the above mentioned strategies:

- MOTRep: MOTR without the extreme point step;
- MOTRsc: MOTR without the scalarization step;
- MOTRmiddle: MOTR using a different strategy than the one described in Algorithm 3 to select the point where to solve the scalarization problem.

In MOTRmiddle, points are sorted according to the value of the objective function component under analysis and an average gap is computed for each point by considering the distances to the two closest points to it, for the objective function component under analysis. The average gaps of the initial and final points of the sorted set of points are equal to the distance of these points to the next or previous one, respectively. The point with the largest average gap, holding a scalarization step trust-region radius larger than or equal to the minimum value allowed for it, is the one selected.

A second goal of the numerical section was to compare the performance of MOTR against other derivative-based multiobjective optimization solvers that intrinsically attempt to generate approximations to the complete Pareto front of a multiobjective optimization problem. With this purpose, MOSQP [15] was selected.

All codes were implemented in MATLAB (version R2021b, was considered). The minimization subproblems of MOTR, defined at the extreme point and scalarization steps, were solved with the MATLAB function `fmincon.m`.

MOTR was described and analyzed for unconstrained optimization. However, the algorithmic description can be easily adapted to incorporate bound constraints, by adding these constraints to the subproblems to be solved. Notice that for convex feasible regions, like is the case of bound constraints, the middle point, computed using Algorithm 3, will remain feasible. Thus, as test set, we considered 54 twice continuously differentiable bound constrained multiobjective optimization problems, available at:

`https://docentes.fct.unl.pt/algb/pages/problems-collections`,

with a number of variables between 1 and 30, and involving 2 or 3 objective function components. A complete list of the problem dimensions can be found in Table 1.

MOTR was run with the parameters $\mu_1 = 0.5$, $\mu_2 = 2$, $\eta_{ep}^1 = \eta_{sc}^1 = 0.001$, $\eta_{ep}^2 = \eta_{sc}^2 = 0.9$, $\Delta_{ep}^{init} = (1, \ldots, 1)^\top \in \mathbb{R}^q$, and $\Delta_{sc}^{init} = 1$. Regarding the update of the trust-region radius at very successful iterations, this was only increased if it limited the progress of the algorithm, meaning that the boundary of the trust-region was reached. In this case, a maximum value of $\Delta^{max} = \|u - l\|/2$ is allowed, where u and l represent the upper and lower bounds of the problem variables. The algorithm was always

Table 1 The set of problems considered in the numerical experiments

Problem	n	q	Problem	n	q	Problem	n	q
BK1	2	2	CL1	4	2	Deb41	2	2
Deb513	2	2	Deb521b	2	2	DG01	1	2
DPAM1	10	2	DTLZ1	7	3	DTLZ1n2	2	2
DTLZ2	12	3	DTLZ2n2	2	2	DTLZ3	12	3
DTLZ3n2	2	2	DTLZ4	12	3	DTLZ4n2	2	2
DTLZ6	22	3	DTLZ6n2	2	2	ex005	2	2
Far1	2	2	Fonseca	2	2	IKK1	2	3
IM1	2	2	Jin1	2	2	Jin3	2	2
L2ZDT2	30	2	L3ZDT2	30	2	lovison1	2	2
lovison2	2	2	lovison3	2	2	lovison4	2	2
lovison5	3	3	lovison6	3	3	LRS1	2	2
MHHM1	1	3	MHHM2	2	3	MLF1	1	2
MLF2	2	2	MOP1	1	2	MOP2	4	2
MOP3	2	2	MOP5	2	3	MOP6	2	2
MOP7	2	3	SK1	1	2	SK2	4	2
SP1	2	2	SSFYY1	2	2	SSFYY2	1	2
TKLY1	4	2	VFM1	2	3	VU1	2	2
VU2	2	2	ZDT2	30	2	ZLT1	10	3

For each problem, n represents the number of variables and q is the number of components of the objective function

initialized with a singleton, namely the centroid of the feasible region. As stopping criteria, minimum trust-region radius $\Delta_{ep}^{min} = \Delta_{sc}^{min} = 10^{-5}$ were allowed, in the case of Δ_{ep} componentwise. Three different budgets were considered in terms of function evaluations, namely 500, 5000, and 20000. For each problem, the approximation to the Pareto front generated by each algorithm corresponds to all the current nondominated points, stored in the list L.

4.1 Performance assessment and metrics

As performance tool, we considered the performance profiles proposed by Dolan and Moré [10], which allow to simultaneously assess the numerical performance of different solvers, for different metrics. The performance of solver $s \in S$ on a given set of problems P is represented by a cumulative function

$$\rho_s(\tau) = \frac{1}{|P|} \left|\{p \in P : r_{p,s} \leq \tau\}\right|,$$

where $\tau \geq 1$ and the performance ratio is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}.$$

Here $t_{p,s}$ represents the value of the selected metric, obtained by solver $s \in S$ when solving problem $p \in P$. Larger values of $\rho_s(\tau)$ indicate a better numerical performance of solver s. In particular, the solver with the largest value of $\rho_s(1)$ is the most efficient. On the other hand, the solver with the largest value of $\rho_s(\tau)$ for large values of τ is the most robust.

Selecting a single metric to compare the performance of multiobjective optimization solvers is always reductive, since advantages and disadvantages can be pointed to each of these indicators. Considering that a good multiobjective optimization solver should be able to generate a large percentage of nondominated points and should also be able to capture the extent of the Pareto front of the multiobjective optimization problem, we decided to consider four metrics that attempt to quantify these features, namely purity, hypervolume and the spread metrics Γ and Δ.

Purity measures the percentage of nondominated points generated by a given solver

$$\bar{t}_{p,s} = Pur_{p,s} = \frac{|F_{p,s} \cap F_p|}{|F_{p,s}|},$$

where $F_{p,s}$ represents the approximation to the Pareto front of problem p computed by solver s and F_p is a reference Pareto front for problem p, computed by considering the union of the Pareto approximations corresponding to all solvers, $\cup_{s \in S} F_{p,s}$, and discarding from it all the dominated points [6].

Hypervolume [32], additionally to nondominance, attempts to capture spread, by measuring the volume of the region dominated by the current approximation to the Pareto front and a reference point $U_p \in \mathbb{R}^q$, that is dominated by all points belonging to the different approximations computed for the Pareto front of problem $p \in P$ by all solvers tested. Mathematically, it can be formalized as:

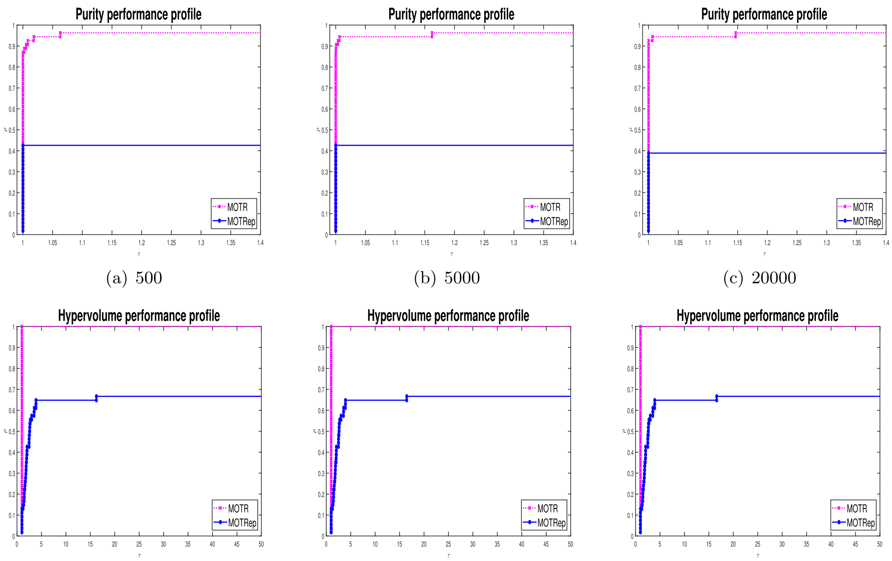
$$\bar{t}_{p,s} = HV_{p,s} = Vol\{y \in \mathbb{R}^q \mid y \leq U_p \wedge \exists x \in F_{p,s} : x \leq y\} = Vol\left(\bigcup_{x \in F_{p,s}} [x, U_p]\right),$$

where $Vol(.)$ denotes the Lebesgue measure of a q-dimensional set of points and $[x, U_p]$ denotes the interval box with lower corner x and upper corner U_p.

For computing the performance profiles, for purity and hypervolume metrics, since larger values indicate a better performance, the inverse value of each one of the metrics was used ($t_{p,s} = 1/\bar{t}_{p,s}$).

Finally, to directly assess spread across the Pareto front, two additionally metrics were considered: the Γ metric, that measures the size of the largest gap in the computed approximation to the Pareto front, and the Δ metric, that assesses how uniformly the nondominated points are distributed along the generated approximation. In a simplified way, consider that solver $s \in S$ has computed, for problem $p \in P$, an approximated Pareto front with points $y_1, y_2, \ldots, y_N$, to which we add the so-called extreme points, y_0 and y_{N+1}, corresponding to the points with the best and worst values for each objective function component. Then

$$\Gamma_{p,s} = \max_{j \in \{1,\ldots,q\}} \left(\max_{i \in \{0,\ldots,N\}} \{\delta_{j,i}\} \right), \tag{15}$$



(a) 500

(b) 5000

(c) 20000

(d) 500

(e) 5000

(f) 20000

Fig. 1 Comparing MOTR and MOTRep based on performance profiles of purity and hypervolume metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed

where $\delta_{j,i} = f_j(y_{i+1}) - f_j(y_i)$, assuming that the objective function values have been sorted by increasing order for each objective function component j. Metric Δ [9] is computed by:

$$\Delta_{p,s} = \max_{j\in\{1,\ldots,q\}} \left(\frac{\delta_{j,0} + \delta_{j,N} + \sum_{i=1}^{N-1} |\delta_{j,i} - \bar{\delta}_j|}{\delta_{j,0} + \delta_{j,N} + (N-1)\bar{\delta}_j} \right), \tag{16}$$

where $\bar{\delta}_j$, for $j = 1, \ldots, q$, represents the average of the distances $\delta_{j,i}, i = 1, \ldots, N-1$.

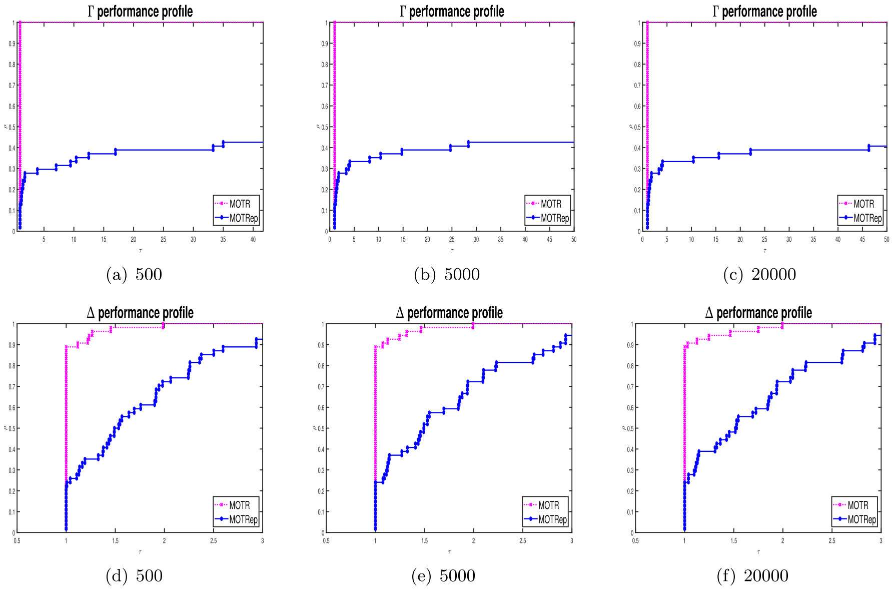
4.2 Adequacy of the algorithmic structure of MOTR

Performance profiles comparing MOTR and MOTRep, a version of MOTR without performing the extreme point step, can be found in Figs. 1 and 2.

Figures 3 and 4 report the comparison between MOTR and MOTRsc, the latter corresponding to a version of MOTR omitting the scalarization step.

Finally, MOTR numerical performance is assessed against MOTRmiddle, where the use of Algorithm 3 is replaced by the strategy described in Sect. 4 to compute the point where to solve the scalarization problems. Figures 5 and 6 report the obtained results.

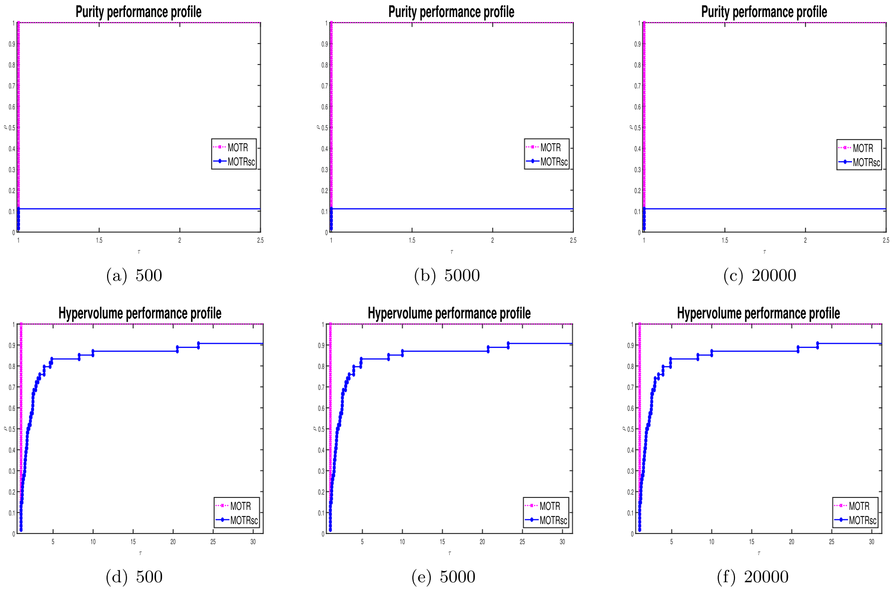
It is clear the advantage of MOTR over each one of its variants, both in terms of efficiency and robustness, for each one of the metrics considered, independently of the budget of function evaluations allowed. The exception appears in the results for



(a) 500 (b) 5000 (c) 20000

(d) 500 (e) 5000 (f) 20000

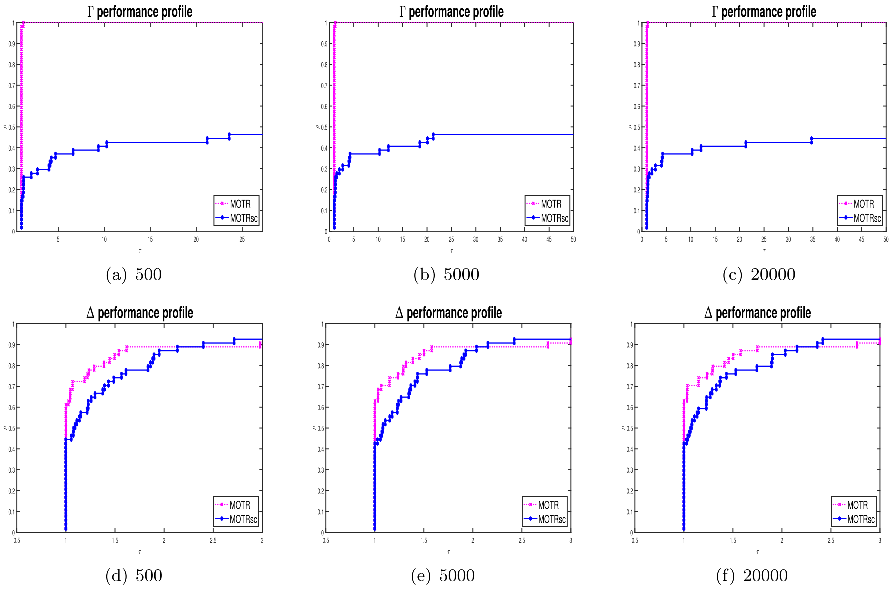
Fig. 2 Comparing MOTR and MOTRep based on performance profiles of Γ and Δ metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed



(a) 500 (b) 5000 (c) 20000

(d) 500 (e) 5000 (f) 20000

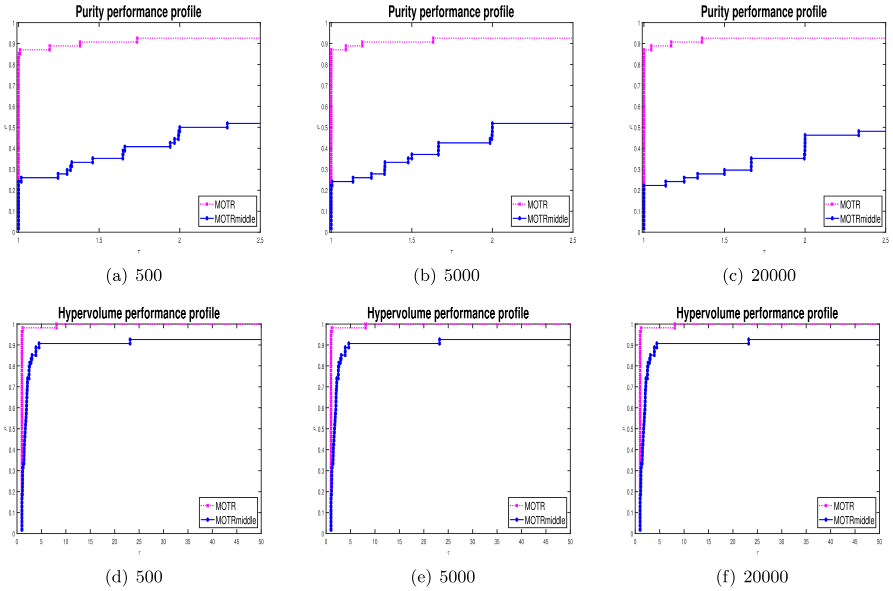
Fig. 3 Comparing MOTR and MOTRsc based on performance profiles of purity and hypervolume metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed



(a) 500 (b) 5000 (c) 20000

(d) 500 (e) 5000 (f) 20000

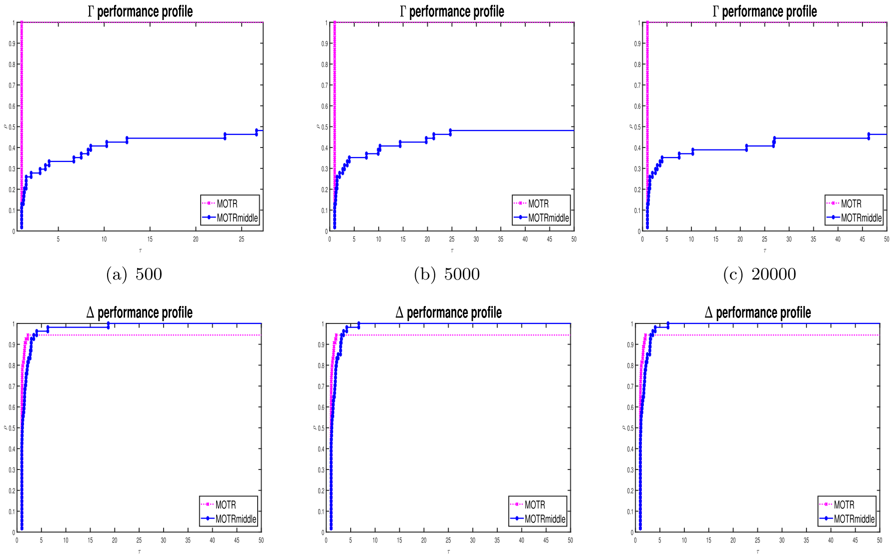
Fig. 4 Comparing MOTR and MOTRsc based on performance profiles of Γ and Δ metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed



(a) 500 (b) 5000 (c) 20000

(d) 500 (e) 5000 (f) 20000

Fig. 5 Comparing MOTR and MOTRmiddle based on performance profiles of purity and hypervolume metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed



(a) 500 (b) 5000 (c) 20000

(d) 500 (e) 5000 (f) 20000

Fig. 6 Comparing MOTR and MOTRmiddle based on performance profiles of Γ and Δ metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed

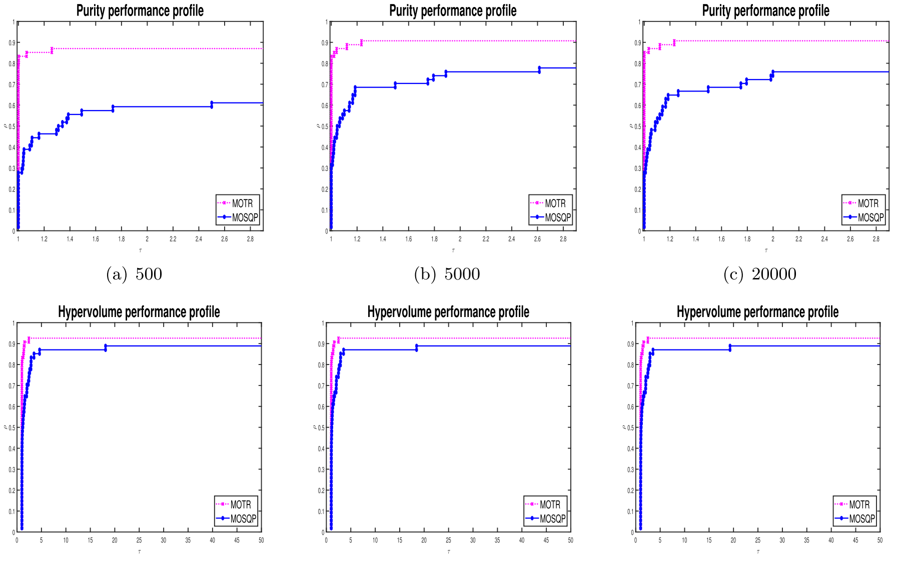
the Δ metric, when comparing MOTR with MOTRsc or MOTRmiddle, where MOTR continues to present a better performance in terms of efficiency, but the results are comparable for robustness.

Although, the clear differences for the remaining metrics and for each one of the variants tested, allow us to conclude that indeed all the three strategies incorporated in the algorithmic structure of MOTR are essential for the good numerical performance of the solver.

4.3 Comparing MOTR with MOSQP

MOSQP was proposed in [15], incorporating in its algorithmic structure strategies to compute approximations to the complete Pareto front of a given multiobjective optimization problem. The solver also keeps a list of points that is updated at each iteration by solving single-objective constrained optimization problems derived as SQP problems. In [15], the authors compared the solver against a classical scalarization approach for biobjective problems and also genetic algorithms. The numerical results reported establish the superiority of MOSQP over the remaining solvers tested.

A MATLAB implementation of MOSQP is distributed by the authors, providing different algorithmic choices. We selected MOSQP $(H = (I, \nabla^2 f), line)$, which corresponds to a line initialization strategy, by computing 200 initial points evenly spaced in the line segment joining the lower and upper bounds of the variables, and where the identity matrix and the true Hessians are used in the second and third algorithmic stages, respectively [15]. This version is reported in [15] as the one that



(a) 500 (b) 5000 (c) 20000

(d) 500 (e) 5000 (f) 20000

Fig. 7 Comparing MOTR and MOSQP based on performance profiles of purity and hypervolume metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed

presents the best computational performance. In regard to stopping criteria, we kept all the default values, but tried the three different budgets of function evaluations. Again, for each problem the approximation to the Pareto front generated by each one of the solvers corresponds to all current nondominated points, stored in the corresponding lists.

Figures 7 and 8 report the performance profiles obtained for the two solvers for purity, hypervolume, and the spread metrics.

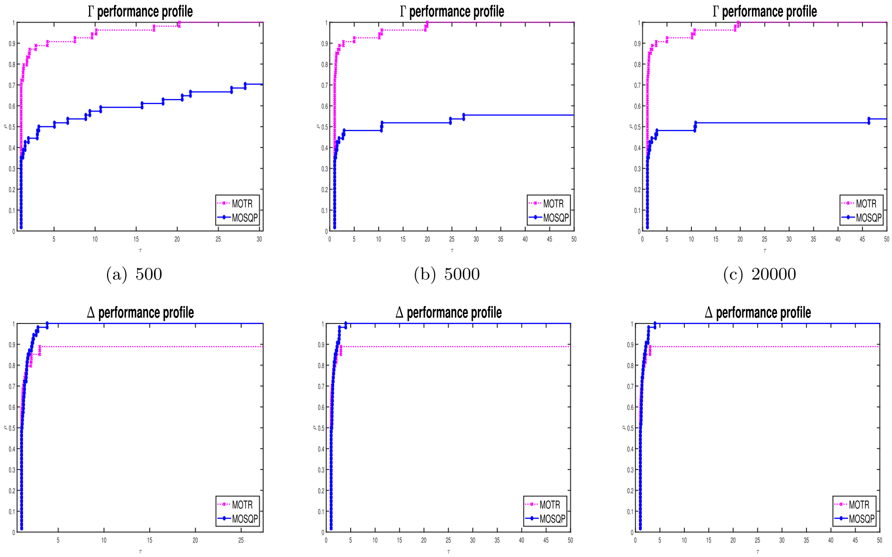
MOTR is clearly competitive, with remarkably good results in terms of efficiency for purity, hypervolume, and Γ. Regarding the uniformity of the distribution of points across the approximation to the Pareto front, MOSQP presents a better performance. These conclusions hold, independently of the budget of function evaluations considered.

Table 2 reports the number of nondominated points obtained by each solver, for a maximum budget of 5000 function evaluations, corroborating the results already reported for the purity metric.

Figures 9 and 10 illustrate the final approximations to the Pareto fronts obtained by MOTR and MOSQP on two biobjective and two triobjective problems, respectively.

5 Conclusions

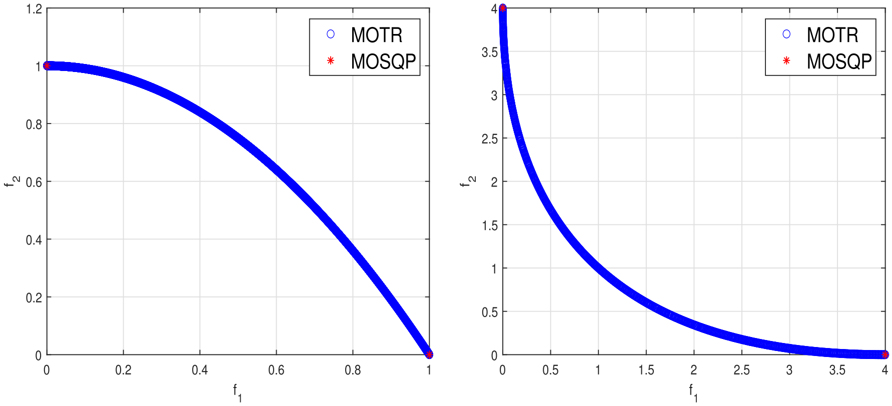
In this work, we proposed a new algorithm, based on a trust-region approach, to compute approximations to the complete Pareto front of multiobjective optimization problems. The algorithmic structure is organized in two main steps: the extreme point



(a) 500 (b) 5000 (c) 20000

(d) 500 (e) 5000 (f) 20000

Fig. 8 Comparing MOTR and MOSQP based on performance profiles of Γ and Δ metrics. Budgets of 500, 5000, and 20000 function evaluations were allowed



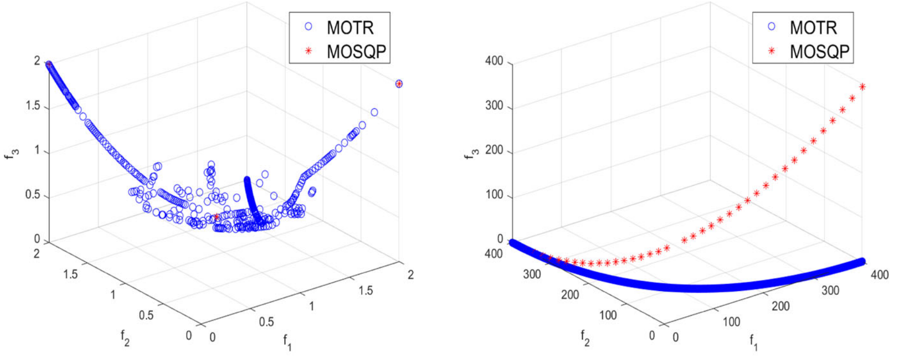
(a) Problem: ZDT2 (b) Problem: MOP1

Fig. 9 Approximations to the Pareto fronts of problems ZDT2 and MOP1, obtained by solvers MOTR and MOSQP, for a budget of 5000 function evaluations

and the scalarization steps, that are alternately performed. As previously mentioned, in the extreme point step, the algorithm tries to reach the extreme points of the Pareto front. On the other hand, in the scalarization step, the focus is on the large gaps in the Pareto front, attempting to reduce them. With this purpose, a new strategy, based on the computation of middle points, was used to select the points to be explored in the scalarization step, by solving adequate scalarization problems.

Table 2 Number of nondominated points in the final approximation of the Pareto front, generated for each problem by MOTR and MOSQP, considering a budget of 5000 function evaluations

Problem	MOTR	MOSQP	Problem	MOTR	MOSQP	Problem	MOTR	MOSQP
BK1	5000	66	CL1	4827	89	Deb41	2197	64
Deb513	1	2	Deb521b	4477	1	DG01	77	25
DPAM1	2496	7	DTLZ1	4999	1	DTLZ1n2	4998	0
DTLZ2	4991	16	DTLZ2n2	4989	16	DTLZ3	4989	1
DTLZ3n2	4988	2	DTLZ4	1	8	DTLZ4n2	1	6
DTLZ6	950	1	DTLZ6n2	2497	1	ex005	4976	67
Far1	1726	52	Fonseca	4996	0	IKK1	3922	36
IM1	4982	90	Jin1	4997	87	Jin3	4834	2
L2ZDT2	0	2	L3ZDT2	752	1	lovison1	4942	93
lovison2	1	8	lovison3	4871	97	lovison4	4373	81
lovison5	412	18	lovison6	269	9	LRS1	4998	1
MHHM1	3936	8	MHHM2	4994	0	MLF1	0	17
MLF2	4938	34	MOP1	5000	0	MOP2	5000	24
MOP3	1610	19	MOP5	2535	12	MOP6	1	2
MOP7	4302	1	SK1	3960	88	SK2	1632	6
SP1	4952	59	SSFYY1	5000	1	SSFYY2	2502	1
TKLY1	595	68	VFM1	4997	49	VU1	0	90
VU2	2498	24	ZDT2	4936	2	ZLT1	383	1



(a) Problem: ZLT1

(b) Problem: IKK1

Fig. 10 Approximations to the Pareto fronts of problems ZLT1 and IKK1, obtained by solvers MOTR and MOSQP, for a budget of 5000 function evaluations

Convergence was analyzed for linked sequences of points generated by MOTR, establishing that any limit point of a linked sequence is a Pareto critical point. As illustrated in the numerical experiments, the existence of each one of these steps is essential for the good numerical performance of the solver, which is quite competitive against MOSQP, a reference solver for multiobjective optimization, when the goal is to compute approximations to complete Pareto fronts.

Acknowledgements This work was funded by national funds through FCT - Fundação para a Ciência e a Tecnologia I.P., under the scope of projects PTDC/MAT-APL/28400/2017, UIDP/00297/2020, and UIDB/00297/2020 (Center for Mathematics and Applications). The work of the first author was additionally supported by the scholarship 2020.08249.BD, also granted by FCT - Fundação para a Ciência e a Tecnologia I.P..

Funding Open access funding provided by FCT|FCCN (b-on).

Data availability statement The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Afshari, H., Hare, W., Tesfamariam, S.: Constrained multi-objective optimization algorithms: review and comparison with application in reinforced concrete structures. Appl. Soft Comput. **83**, 105631 (2019)
2. Anagnostopoulos, K., Mamanis, G.: Multiobjective evolutionary algorithms for complex portfolio optimization problems. Comput. Manag. Sci. **8**, 259–279 (2011)
3. Ansary, M.A.T., Panda, G.: A globally convergent SQCQP method for multiobjective optimization problems. SIAM J. Optim. **31**, 91–113 (2021)
4. Carrizo, G.A., Lotito, P.A., Maciel, M.C.: Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem. Math. Program. **159**, 339–369 (2016)
5. Conn, A.R., Gould, N., Toint, Ph.: Trust-Region Methods. MOS-SIAM Ser. Optim. SIAM, Philadelphia (2000)
6. Custódio, A.L., Madeira, J.F.A., Vaz, A.I.F., Vicente, L.N.: Direct multisearch for multiobjective optimization. SIAM J. Optim. **21**, 1109–1140 (2011)
7. Das, I., Dennis, J.E.: A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct. Multidiscip. Optim. **14**, 63–69 (1997)
8. Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J. Optim. **8**, 631–657 (1998)
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE T. Evolut. Comput. **6**, 182–197 (2002)
10. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
11. Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Vector Optimization, Springer, Heidelberg (2008)
12. Emmerich, M.T.M., Deutz, A.H.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat. Comput. **17**, 585–609 (2018)
13. Fliege, J., Drummond, L.M.G., Svaiter, B.F.: Newton's method for multiobjective optimization. SIAM J. Optim. **20**, 602–626 (2009)

14. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. **51**, 479–494 (2000)
15. Fliege, J., Vaz, A.I.F.: A method for constrained multiobjective optimization based on SQP techniques. SIAM J. Optim. **26**, 2091–2119 (2016)
16. Fonseca, C.M., Paquete, L., López-Ibánez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: Proceedings of the 2006 Congress on Evolutionary Computation (CEC'06), pp. 1157–1163. IEEE (2006)
17. Fukuda, E.H., Drummond, L.M.G.: A survey on multiobjective descent methods. Pesq. Oper. **34**, 585–620 (2014)
18. Griva, I., Nash, S.G., Sofer, A.: Linear and Nonlinear Optimization, 2nd edn. SIAM, Philadelphia (2009)
19. Liuzzi, G., Lucidi, S., Rinaldi, F.: A derivative-free approach to constrained multiobjective nonsmooth optimization. SIAM J. Optim. **26**, 2744–2774 (2016)
20. Luc, D.T., Phong, T.Q., Volle, M.: Scalarizing functions for generating the weakly efficient solution set in convex multiobjective problems. SIAM J. Optim. **15**, 987–1001 (2005)
21. Miettinen, K.: Nonlinear Multiobjective Optimization. International Series in Operations Research & Management Science, Springer US, New York (1998)
22. Moré, J.J.: Recent developments in algorithms and software for trust region methods. In: Mathematical Programming The State of the Art: Bonn 1982, pp. 258–287. Springer, Berlin, Heidelberg (1983)
23. Morovati, V., Basirzadeh, H., Pourkarimi, L.: Quasi-Newton methods for multiobjective optimization problems. 4OR **16**, 261–294 (2018)
24. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer (2006)
25. Potrebko, P.S., Fiege, J., Biagioli, M., Poleszczuk, J.: Investigating multi-objective fluence and beam orientation IMRT optimization. Phys. Med. Biol. **62**, 5228–5244 (2017)
26. Qu, S., Goh, M., Liang, B.: Trust region methods for solving multiobjective optimisation. Optim. Methods Softw. **28**, 796–811 (2013)
27. Ramirez, V.A., Sottosanto, G.N.: Nonmonotone trust region algorithm for solving the unconstrained multiobjective optimization problems. Comput. Optim. Appl. **81**, 769–788 (2022)
28. Roy, P.C., Blank, J., Hussein, R., Deb, K.: Trust-region based algorithms with low budget for multiobjective optimization. In: GECCO18: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 195–196 (2018)
29. Tapia, M.G.C., Coello, C.A.C.: Applications of multi-objective evolutionary algorithms in economics and finance: a survey. In: 2007 IEEE congress on evolutionary computation, pp. 532–539 (2007)
30. Thomann, J., Eichfelder, G.: A trust-region algorithm for heterogeneous multiobjective optimization. SIAM J. Optim. **29**, 1017–1047 (2019)
31. Villacorta, K.D.V., Oliveira, P.R., Soubeyran, A.: A trust-region method for unconstrained multiobjective problems with applications in satisficing processes. J. Optim. Theory Appl. **160**, 865–889 (2014)
32. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE T. Evolut. Comput. **7**, 117–132 (2003)