Complexity bound of trust-region methods for convex smooth unconstrained multiobjective optimization

R. Garmanjani *

June 9, 2022

Abstract

In this paper, we analyze the worst-case complexity of trust-region methods for solving unconstrained smooth multiobjective optimization problems. We particularly focus on the method proposed by Villacorta et al. [J. Optim. Theory Appl., 160:865–889]. When the component functions are convex (respectively strongly convex), we will derive a complexity bound of $\mathcal{O}(\epsilon^{-1})$ (respectively $\mathcal{O}(\log \epsilon^{-1})$) for driving some criticality measure below some given positive ϵ. The derived complexity bounds recover those of classical trust-region methods for solving (strongly) convex smooth unconstrained single-objective problems.
Keywords: trust-region methods, multiobjective optimization, worst-case complexity, convex smooth unconstrained

AMS Subject Classification (2020): 90C25, 90C29, 90C30, 90C60

1 Introduction

In multiobjective optimization, in contrast to single-objective optimization, one needs to minimize more than one possibly conflicting objectives at the same time. This kind of optimization problems appears in many applications in science, engineering, finance, and economic, among others. There are various approaches and a vast number of methods in literature which have been devoted to solving multiobjective optimization problems. An extensively studied approach for solving a multiobjective optimization problem is the so-called *scalarization* approach in which a single (and sometimes more than one) objective, made by, for instance, a weighted sum of all the objectives, is minimized. The weighting parameters are set by the optimizer based on his/her preference in advance or are chosen adaptively as the algorithm proceeds. Further details on such approach can been found in [1, 2, 3]. Another approach is based on the extension of single-objective optimization algorithms to multiobjective setting, and tries to find a descent direction for all the component functions at the same time, by solving a subproblem at each iteration. These methods do not use a set of parameters in advance for converting the problem to a single-objective optimization. They include steepest descent [4], Newton [5], trust-region

*Center for Mathematics and Applications (NovaMath), FCT NOVA, 2829-516, Caparica, Portugal (`r.garmanjani@fct.unl.pt`). Support for the author was provided by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) under the projects PTDC/MAT-APL/28400/2017, UIDP/MAT/00297/2020, and UIDB/MAT/00297/2020.

methods [6, 7, 8, 9], among others. Further information on such methods can be found in the survey [10] and the references therein.

In this paper, we consider unconstrained multiobjective optimization problems of the form

$$\min_{x\in\mathbb{R}^n} F(x) = (f_1(x), \ldots, f_m(x)), \tag{1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in I := \{1, \ldots, m\}$, is a smooth function. For this class of optimization problems, under the convexity assumption of the component functions, we are interested in analyzing the worst-case complexity (WCC) of a trust-region method for driving the criticality measure below some given positive threshold. The WCC of an algorithm measures the cost of running the algorithm, which is expressed in terms of number of iterations or function evaluations, in the worst-case scenario when starting from any initial point and arriving at a point for which the stationary measure is small enough.

Hereafter, when we refer to nonconvex, convex, or strongly convex case, we mean all the component functions in (1) are nonconvex, convex, or strongly convex, respectively.

By setting $m = 1$ in (1), we will have an unconstrained single-objective optimization problem. The WCC analysis of optimization algorithms, when the goal is to find a point at which the norm of the gradient becomes less than some given positive threshold, has been the subject of many research studies in single-objective optimization of which we will review the ones more related to the current study. For nonconvex smooth unconstrained optimization problems, a WCC bound of $\mathcal{O}(\epsilon^{-2})$ has been derived for steepest descent methods [11] [see Example 1.2.3] and trust-region methods [12]. The same complexity bound has been shown to hold for adaptive regularization with cubic (ARC) [13]. A complexity bound of $\mathcal{O}(n^2\epsilon^{-2})$ has been obtained in [14] for derivative-free trust-region methods. Assuming the convexity of the objective function, a complexity bound of $\mathcal{O}(\epsilon^{-1})$ has been derived in [15] for nonlinear stepsize control (NSC) framework [16], which is a generalized framework for ARC and trust-region methods. Under strong convexity condition, a WCC bound of $\mathcal{O}(\log(\epsilon^{-1}))$ has been derived in [17] for NSC framework.

Despite ubiquity of the WCC analysis of algorithms for single-objective optimization, there are rather few studies addressing the complexity of multiobjective optimization algorithms. The complexity of gradient descent for multiobjective optimization has been analyzed in [18]. A complexity bound of $\mathcal{O}(\epsilon^{-2})$ has been established for the nonconvex case for bringing the Pareto criticality measure below ϵ. The authors have further derived a complexity bound of $\mathcal{O}(\epsilon^{-1})$ and $\mathcal{O}(\log(\epsilon^{-1}))$ for the weighted sum of the component functions. In [19], complexity bounds have been derived for gradient descent multiobjective optimization on Riemannian manifolds. Complexity bounds of $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-1})$ have been derived for the nonconvex and convex case, respectively. In [20], under the assumption of β–Hölder continuity of the p–th order derivative of component functions, the authors have developed a high-order regularization mutliobjective method along with a complexity bound of $\mathcal{O}(\epsilon^{-\frac{p+\beta}{p+\beta-1}})$, which when $p = \beta = 1$ (i.e. when f_is are continuously differentiable with Lipschitz continuous gradient) leads to the complexity bound of $\mathcal{O}(\epsilon^{-2})$. The complexity of the Direct Multisearch (DMS) algorithm and a particular version of it has been analyzed in [21], and a complexity bound of $\mathcal{O}(\epsilon^{-2})$ has been derived for the latter case. In [22], it has been shown that a stochastic multi-gradient method enjoys complexity bounds of $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-1})$ under convexity and strong convexity assumptions, respectively.

In [9], a trust-region multiobjective method, which is called TRMP, has been proposed for solving (1). The WCC of TRMP algorithm for the nonconvex case (i.e. when the component functions in (1) are nonconvex) has been analyzed in [23]. The authors have derived a complexity bound of $\mathcal{O}(\epsilon^{-2})$ for the method. To the best of our knowledge, when the component functions in (1) are (strongly) convex, no complexity bound has been derived for trust-region methods and more specifically for TRMP algorithm. In this paper, we aim to address this gap. Under convexity (resp. strong convexity) assumption of the component functions in (1), we derive a complexity bound of $\mathcal{O}(\epsilon^{-1})$ (resp. $\mathcal{O}(\log \epsilon^{-1})$) for finding an ϵ–Pareto critical point (see Definition 2.2). For the convex case, the bound matches the one derived in [19] for steepest descent multiobjective optimization on Riemannian manifolds. Furthermore, when $m = 1$, this recovers the complexity bound of driving the norm of gradient below ϵ, which has been derived for the trust-region methods under the NSC framework in [15] and [17] for convex and strongly convex case, respectively. Additionally, as a byproduct of our WCC analysis, we obtain the WCC bounds also for trust-region methods for a class of *structured nonsmooth* (strongly) convex single-objective problems (i.e. pointwise maximum of (strongly) convex functions). We should further add that, although we use a minimax approach for converting the original problem to a single-objective optimization problem, the resulting function is no longer smooth. Therefore, the WCC result derived here is not resulted in a trivial way by applying the techniques used in [15] for establishing WCC bound of single-objective optimization problems, which rely on the smoothness of the objective function.

The rest of the paper is organized as follows. In Section 2 we present the TRMP algorithm in its original form as given in [9], and introduce some preliminary results. The WCC analysis of the method for the convex case is established in Section 3. Finally, we will draw some conclusions and mention some future lines of research in Section 5.

Notation. In this paper, we mostly follow the related notations used in [9]. For simplicity, we assume all the norms are Euclidean norm. Notation $|\cdot|$ represents the cardinality of a set. For set $X \subset \mathbb{R}^n$, $\text{conv}(X)$ denotes its convex hull. The subgradient of function $f : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ with ∂f. For $q \in \mathbb{N}$, we define the nonegative orthant (which is also called *Pareto cone*) $\mathbb{R}^q_+ =: \{(u_1, \dots, u_q) \in \mathbb{R}^q | u_i \geq 0\}$, and the positive orthant $\mathbb{R}^q_{++} =: \{(u_1, \dots, u_q) \in \mathbb{R}^q | u_i > 0\}$. Given $u, v \in \mathbb{R}^q$, we say $u \preceq v$ (resp. $u \prec v$) iff $v - u \in \mathbb{R}^q_+$ (resp. $v - u \in \mathbb{R}^q_{++}$). We denote the set of all symmetric matrices of size $n \times n$ belonging to $\mathbb{R}^{n\times n}$ with $\mathbb{S}^{n\times n}$.

2 TRMP algorithm and some preliminary results

Throughout the paper, we will use the following quantity as the *criticality measure*:

$$\omega(x) \;:=\; -\min_{\|d\|\leq 1} \left(\max_{i\in I} \nabla f_i(x)^\top d \right). \tag{2}$$

We notice that for the single-objective case (i.e. $m = 1$ in (1)), we have $\omega(x) = \|\nabla F(x)\|$.

Definition 2.1 *[4, 5] Let $x_* \in \mathbb{R}^n$. We call x_**

(i) a Pareto optimal if there does not exists $y \in \mathbb{R}^n$ such that

$$F(y) \;\preceq\; F(x_*) \text{ and } F(y) \;\neq\; F(x_*);$$

(ii) a weakly Pareto optimal if there does not exists $y \in \mathbb{R}^n$ such that

$$F(y) \;\prec\; F(x_*);$$

(iii) a Pareto critical for F if

$$\text{range}(JF(x)) \cap (-\mathbb{R}^m_{++}) \;=\; \emptyset, \tag{3}$$

where range$(JF(x))$ *denotes the linear subspace generated by the columns of Jacobian of F at x, and $-\mathbb{R}^m_{++} = \{-u | u \in \mathbb{R}^m_{++}\}$.*

Let

$$\mathcal{D}(x) \;=\; \{d(x) : d(x) \text{ is a solution of } (2)\}.$$

The following lemma lists some properties of ω and its relation with Pareto criticality.

Lemma 2.1 *[4]*

(i) $\omega(x) \geq 0$, for all $x \in \mathbb{R}^n$;

(ii) if x is Pareto critical of (1), then $0 \in \mathcal{D}(x)$ and $\omega(x) = 0$;

(iii) if x is not Pareto critical of (1), then $\omega(x) > 0$ and for any $d \in \mathcal{D}(x)$ we have

$$\nabla f_j(x)^\top d \;\leq\; \max_{i \in I} \left\{ \nabla f_i(x)^\top d \right\} \;<\; 0, \quad \forall j \in I,$$

i.e., d is a descent direction of (1);

(iv) the application $x \mapsto \omega(x)$ is continuous.

Definition 2.2 *[24] Let $x \in \mathbb{R}^n$. We call x an ϵ–Pareto critical point if*

$$\omega(x) \;\leq\; \epsilon. \tag{4}$$

The following lemma, which is a slightly modified version of Theorem 3.1 in [5], establishes relationship between (weak) Pareto optimality and Pareto criticality.

Lemma 2.2 *[5, Theorem 3.1] Assume the component function f_i, for each $i \in I$, is continuously differentiable on $\mathbb{R}^n$.*

1. *If $\bar{x}$ is locally weak Pareto optimal point, then $\bar{x}$ is a critical point for F.*

2. *If f_i, for each $i \in I$, is convex and $\bar{x} \in \mathbb{R}^n$ is Pareto critical for F, then $\bar{x}$ is weak Pareto optimal.*

3. *If $\nabla^2 f_i(x)$ is positive definite for all $i \in I$ and $x \in \mathbb{R}^n$, and if $\bar{x} \in \mathbb{R}^n$ is Pareto critical for F, then $\bar{x}$ is Pareto optimal.*

In view of Lemma 2.2, under the convexity assumption of the component functions, any weak Pareto optimal is Pareto critical for F and vice versa.

The following lemma states that the subgradient of a *pointwise maximum* of m closed and convex functions at a point is equal to the convex hull of the subgradients of those functions that are active at that point.

Lemma 2.3 *[11, Lemma 3.1.10] Let function $f_i(x)$, for all $i \in I$, be closed and convex. Then function*

$$\phi(x) := \max_{i \in I} f_i(x) \tag{5}$$

is also closed and convex. For any $x \in \operatorname{int}(\operatorname{dom} \phi) = \cap_{i=1}^m \operatorname{int}(\operatorname{dom} f_i)$ we have

$$\partial \phi(x) = \operatorname{conv}\{\partial f_i(x) : i \in \mathcal{I}(x)\} \tag{6}$$

where $\mathcal{I}(x) = \{i : f_i(x) = \phi(x)\}$.

In trust-region methods for minimizing single-objective optimization problems, to find a new point, typically a quadratic model within a region is minimized. If the magnitude of a quotient, which is resulted from dividing the *actual reduction* (i.e. reduction in function value) by the *predicted reduction* (i.e. reduction in model value), is close enough to 1, the new point would then be accepted and the trust-region radius is either retained or increased; otherwise the new point is rejected and the trust-region radius is reduced and the quadratic model is minimized within the new region. Further details on trust-region methods for single-objective optimization can be found in the comprehensive monograph [25].

In the following, we briefly describe TRMP algorithm, as originally proposed in [9], which is a trust-region algorithm for solving (1). We refer the interested reader to [9] for further details on the method and its application. We should mention that there are other trust-region type methods developed for solving multiobjective optimization problems [6, 7].

In TRMP algorithm, which is an extension of the classical trust-region method [25], the idea of solving (1) revolves around substituting it with the following *strict scalarization*:

$$\min_{x \in \mathbb{R}^n} \phi(x) = \max_{i \in I} f_i(x). \tag{7}$$

(We notice that, in view of (6) and as the component functions f_i are assumed to be continuously differentiable, we have

$$\partial \phi(x) = \operatorname{conv}\{\nabla f_i(x) : i \in I(x)\}.) \tag{8}$$

Remark 2.1 *The aforementioned scalarization approach is different from the so-called standard scalarization approach, which is based on transforming the original problem to a single-objective function by a choosing a set of parameters a priori. In the case of the TRMP algorithm, similarly to the methods proposed in [4, 5], these parameters are determined by solving a subproblem (see subproblem* (9) *below) at each iteration of the algorithm.*

The TRMP method uses the following quadratic approximation of function ϕ at point x:

$$m(x, H, d) \ := \ \max_{i \in I} \left\{ f_i(x) + \nabla f_i(x)^\top d \right\} + \frac{1}{2} d^\top H d, \qquad \text{with } H \in \mathbb{S}^{n \times n}.$$

Then, for step calculation, at each iteration of the TRMP algorithm the following subproblem is solved:

$$\begin{aligned} &\min m(x_k, H_k, d), \\ &\text{s.t. } \|d\| \ \leq \ \Delta_k, \end{aligned} \tag{9}$$

where Δ_k is the trust-region radius. Function ϕ and model m are then used to compute the quotient for acceptance of trial point. Finally, the strategy for updating the trust-region is similar to the classical trust-region methods in single-objection optimization. We notice, by setting $m = 1$, we recover the basic trust-region algorithms proposed in [25] for minimization of single-objective problems.

In the continuation, we will use the following notations:

$$m_k(\cdot) \ := \ m\,(x_k, H_k, \cdot)$$

and

$$\mathcal{B}_k \ := \ \{d \in \mathbb{R}^n : \|d\| \leq \Delta_k\}\,.$$

Algorithm 2.1 (TRMP Algorithm [9])

Step 0: Initialization. Choose x_0, H_0, Δ_0, $0 < \eta_1 \leq \eta_2 < 1$, and $0 < \gamma_1 \leq \gamma_2 < 1 < \gamma_3$. Set $k = 0$.

Step 1: Step calculation. Compute a step $d_k \in \mathcal{B}_k$ that "sufficiently reduces" the function m_k.

Step 3: Acceptance of trial point. Compute $\phi(x_k + d_k)$ and

$$\rho_k \ = \ \frac{\phi(x_k) - \phi(x_k + d_k)}{m_k(0) - m_k(d_k)}. \tag{10}$$

If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + d_k$; otherwise define $x_{k+1} = x_k$.

Step 4: Trust-region radius update. Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \gamma_3\Delta_k) & \text{if } \rho_k \geq \eta_2, & [k \text{ very successful}]; \\ [\gamma_2\Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), & [k \text{ successful}]; \\ [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k < \eta_1, & [k \text{ unsuccessful}]. \end{cases} \tag{11}$$

Update H_k to H_{k+1}. Set $k = k + 1$ and go to Step 1.

We denote $\mathcal{S}$ the set of all successful or very successful iterations, i.e.,

$$\mathcal{S} \ = \ \{k \geq 0 : \text{iteration } k \text{ is successful or very successful}\}, \tag{12}$$

and $\mathcal{S}_\ell$ be the set of all such iterations up to iteration ℓ, i.e.,

$$\mathcal{S}_\ell \ = \ \{k \leq \ell : k \in \mathcal{S}\}. \tag{13}$$

We also denote the set of all unsuccessful iterations with $\mathcal{U}$ and the ones up to iteration ℓ with $\mathcal{U}_\ell$. Henceforth, for the sake of simplicity, we will call all successful and very successful iterations as successful iterations.

In the following, we will list several assumptions in a more formal way, which will be required in the continuation for establishing the WCC results.

Assumption 2.1 *For each $i \in I$, the component function f_i is twice continuously differentiable on $\mathbb{R}^n$ and lower bounded.*

Remark 2.2 *For establishing the results in this paper, we need to assume the component functions f_is are just continuously differentiable with Lipschitz continuous gradient rather than being twice continuous differentiable. However, for the sake of conciseness, we stick to the set of assumptions given in [9] in order to avoid proving the results presented there anew under such assumption. In fact, with just some simple modifications in the proofs given in [9], we will be able to show the results there hold under the continuous differentiability of the component functions and the Lipschitz continuity of their gradients.*

Remark 2.3 *As it has been discussed in [9], the assumption on the lower boundedness of all the component functions f_i is not restrictive. As, if unbounded, we can replace* (1) *with the following problem:*

$$\min_{x \in \mathbb{R}^n} F(x) = (\exp(f_1(x)), \ldots, \exp(f_m(x))). \tag{14}$$

Then, one can easily see that all the component functions in (14) *are convex and lower bounded and both problems* (1) *and* (14) *have the same set of Pareto critical points.*

We need to assume that the norm of the Hessian of the component function is bounded from above.

Assumption 2.2 *[9, Assumption 5.1] There exists a positive constant κ_{uFh} such that, for all $x \in \mathbb{R}^n$,*

$$\|\nabla^2 f_i(x)\| \leq \kappa_{uFh}, \quad \text{for all } i \in I.$$

When the gradient of the function f_i is Lipschitz continuous with constant $L_{\nabla f_i}$, then Assumption 2.2 is satisfied with $\kappa_{uFh} = L_{\nabla f_i}$.

We also need to assume that the matrix H_k used for building the model in (9) is uniformly bounded from above.

Assumption 2.3 *[9, Assumption 5.2] The matrix H_k is uniformly bounded, that is, there exists a constant $\kappa_{umh} \geq 1$ such that*

$$\|H_k\| \leq \kappa_{umh} - 1, \quad \text{for all } k.$$

As for the next assumption, we will use the same assumption as [9], which states that the model is *sufficiently reduced* at every iteration. Assumptions 2.1–2.3 are essential in justifying this assumption. We refer the interested reader to [9] for justification on having such assumption.

Assumption 2.4 *[9, Assumption 4.1] For all k,*

$$m_k(0) - m_k(d_k) > \kappa_{dlam}\omega(x_k) \min\left\{\frac{\omega(x_k)}{1 + \|H_k\|}, \Delta_k\right\},$$

for some $\kappa_{dlam} \in (0, 1)$.

The following key lemma, which is a slightly modified version of Proposition 6.2 in [9], gives a lower bound on trust-region radius when the corresponding gradient is lower bounded. This lemma is of central importance in establishing our WCC analysis.

Lemma 2.4 *Let Assumptions 2.1, 2.2, 2.3,and 2.4 hold. Suppose furthermore that there exists a constant $\kappa > 0$ such that $\omega(x_k) \geq \kappa$ for all k. Then*

$$\Delta_k \;\geq\; \kappa_1 \kappa, \quad \textit{for all } k. \tag{15}$$

where

$$\kappa_1 \;=\; \frac{\gamma_1 \kappa_{dlam}(1-\eta_2)}{\kappa_H}, \quad \kappa_H \;=\; \max\{\kappa_{uFh}, \kappa_{umh}\}, \tag{16}$$

and $\kappa_{dlam} \in (0,1)$.

Proof. See the proof of Proposition 6.2 in [9]. ■

The following results states that the number of unsuccessful iterations is bounded by the number of successful iterations. As a result, in order to count the total number of iterations, it suffices to count the number of successful iterations. The proof follows along the lines of the proof of corresponding results given, for instance, in [15].

Theorem 2.1 *Given any $\epsilon \in (0,1)$. Let $\omega(x_k) > \epsilon$ for $k = 0, 1, \cdots, \ell$. Then Algorithm 2.1 takes at most $|\mathcal{U}_\ell|$ unsuccessful iterations, where*

$$|\mathcal{U}_\ell| \;\leq\; -\log_{\gamma_2}(\gamma_3)|\mathcal{S}_\ell| + \log_{\gamma_2} \frac{\kappa_1 \epsilon}{\Delta_0}$$

for having $\|\omega_{\ell+1}\| \leq \epsilon$.

3 WCC of TRMP algorithm under convexity

In this section, we analyze the WCC of TRMP algorithm for minimization of (1). In the following assumption, we formally assume the convexity of the component functions along with the boundedness of the level set of their pointwise maximum function.

Assumption 3.1 *For each $i \in I$, the component function f_i is convex and there exists $D \geq 1$ such that $L(x_0) \subset B(x_*; D)$, where x_* is any global minimizer of ϕ defined in (5).*

We notice, in view of Lemma 2.3, that function ϕ is convex.

The following auxiliary lemma relates the value of ω at a point to the norm of an element of the subgradient of ϕ.

Lemma 3.1 *For any $x \in \mathbb{R}^n$, there exists $g_0 \in \partial\phi(x)$ such that $\omega(x) = \|g_0\|$.*

Proof. If x is Pareto critical then, in view of Lemma 2.1, we have $\omega(x) = 0$, and the result holds with $g_0 = 0$. Suppose x is not a Pareto critical point. Now, the following minimization problem

$$\omega(x) \;=\; -\min_{\|d\| \leq 1} \left(\max_{i \in I} \nabla f_i(x)^\top d \right), \tag{17}$$

can be rewritten as

$$\begin{aligned} \min_{d,t} \quad & t \\ \text{s.t.} \quad & \nabla f_i(x)^\top d \;\leq\; t \quad \forall i \in I, \\ & \|d\| \;\leq\; 1, \end{aligned} \tag{18}$$

with the Lagrangian function

$$\mathcal{L}(d,t,\alpha,\beta) \;=\; t + \sum_{i\in I} \alpha_i(\nabla f_i(x)^\top d - t) + \frac{\beta}{2}\left(\|d\|^2 - 1\right).$$

In view of KKT conditions, we have $\sum_{i\in I} \alpha_i = 1$ and $\sum_{i\in I} \alpha_i \nabla f_i(x) + \beta d = 0$, where $\alpha_i, \beta \geq 0$. Since x is not a Pareto critical point, we have $\beta > 0$. Thus, the dual objective function is given as follows

$$q(\alpha,\beta) \;=\; -\frac{1}{2\beta}\left\|\sum_{i\in I} \alpha_i \nabla f_i(x)\right\|^2 - \frac{\beta}{2}.$$

Hence

$$\begin{aligned} \omega(x) \;=\; -\max_{\alpha,\beta} q(\alpha,\beta) & \qquad\qquad = \min_{\alpha,\beta} \frac{1}{2\beta}\left\|\sum_{i\in I} \alpha_i \nabla f_i(x)\right\|^2 + \frac{\beta}{2}. \\ \text{s.t.} \sum_{i\in I} \alpha_i = 1,\; \alpha_i \geq 0, & \qquad\qquad \text{s.t.} \sum_{i\in I} \alpha_i = 1,\; \alpha_i \geq 0, \end{aligned}$$

Now, from $\nabla_\beta q(\alpha,\beta) = 0$, we obtain $\beta = \|\sum_{i\in I} \alpha_i \nabla f_i(x)\|$. Therefore, we have

$$\begin{aligned} \omega(x) \;=\; \min_{\alpha} & \left\|\sum_{i\in I} \alpha_i \nabla f_i(x)\right\|, \\ \text{s.t.} & \sum_{i\in I} \alpha_i = 1,\; \alpha_i \geq 0, \end{aligned}$$

and as $\partial\phi(x) = \text{conv}\{\nabla f_i(x) : i \in I(x)\}$, the thesis follows. ■

We will need the following lemma which shows that the difference between the function value at any point and optimal function value is of the same order as the value of ω at that point.

Lemma 3.2 *Let Assumption 3.1 hold. Then for any iteration x_k produced by Algorithm 2.1, we have*

$$\Theta_k \;:=\; \phi(x_k) - \phi(x_*) \;\leq\; D\omega(x_k). \tag{19}$$

Proof. Since ϕ is convex, for all $x, y \in \mathbb{R}^n$, we have

$$\phi(y) \;\geq\; \phi(x) + g^\top(y - x),$$

for all $g \in \partial\phi(x)$. In view of Lemma 3.1, we have $\omega(x_k) = \|g_0\|$ for some $g_0 \in \partial\phi(x_k)$. Now, by setting $x = x_k$, $y = x_*$, and using the Cauchy–Schwarz inequality, the desired inequality is resulted. ■

Now, as it was mentioned before, for counting the total number of iterations, we need to count only the number of successful iterations. In the continuation, without loss of generality, we shall assume that the set of successful iterations is not empty. The following theorem shows that the difference between the function value at any point and optimal function value is of the order of the inverse of the number of successful iterations.

Theorem 3.1 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 3.1 hold. Then, by applying Algorithm 2.1, we have*

$$\Theta_\ell \;\le\; \frac{1}{|\mathcal{S}_\ell|\eta_1\kappa_2}, \quad \forall \ell \ge 0,$$

where Θ_ℓ *is given in* (19) *and*

$$\kappa_2 \;=\; \frac{\kappa_{dlam}}{D^2}\min\left\{\frac{1}{1+\kappa_H}, \kappa_1\right\}.$$

Proof. In view of Lemma 3.2, we have $\omega(x_k) \ge \Theta_k/D$. By applying Lemma 2.4 with $\kappa = \Theta_k/D$, we obtain $\Delta_k \ge \kappa_1\Theta_k/D$. Thus, in view of Assumption 2.4, we have

$$\begin{aligned} m_k(0) - m(d_k) \;&\ge\; \kappa_{dlam}\omega(x_k)\min\left\{\frac{\omega(x_k)}{1+\kappa_H}, \Delta_k\right\} \\ &\ge\; \frac{\kappa_{dlam}\Theta_k}{D}\min\left\{\frac{\Theta_k}{(1+\kappa_H)D}, \frac{\kappa_1\Theta_k}{D}\right\} \\ &\ge\; \frac{\kappa_{dlam}\Theta_k^2}{D^2}\min\left\{\frac{1}{1+\kappa_H}, \kappa_1\right\}. \end{aligned}$$

Now, for any $k \in \mathcal{S}$, we have

$$\phi(x_k) - \phi(x_{k+1}) \;\ge\; \eta_1\kappa_2\Theta_k^2, \tag{20}$$

Hence,

$$\Theta_k - \Theta_{k+1} \;\ge\; \eta_1\kappa_2\Theta_k^2, \quad k \in \mathcal{S},$$

which, by dividing both sides by $\Theta_k\Theta_{k+1}$ and considering that $\Theta_{k+1} \ge \Theta_k$ in view of Assumption 2.4, leads to

$$\frac{1}{\Theta_{k+1}} - \frac{1}{\Theta_k} \;\ge\; \eta_1\kappa_2\frac{\Theta_k}{\Theta_{k+1}} \ge \eta_1\kappa_2, \quad k \in \mathcal{S}.$$

By summing up the above inequalities up to ℓ we have,

$$\frac{1}{\Theta_\ell} \;\ge\; \frac{1}{\Theta_0} + |\mathcal{S}_\ell|\eta_1\kappa_2 \ge |\mathcal{S}_\ell|\eta_1\kappa_2 \quad \ell \ge 0.$$

Therefore,

$$\Theta_\ell \;\le\; \frac{1}{|\mathcal{S}_\ell|\eta_1\kappa_2},$$

and the proof is completed. ■

In view of Theorem 2.1 and Theorem 3.1, the following corollary is readily resulted.

Corollary 3.1 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 3.1 hold. Then, for any* $\epsilon \in (0,1)$*, Algorithm 2.1 needs at most* $\mathcal{O}(\epsilon^{-1})$ *iterations for finding a point* $\bar{x} \in \mathbb{R}^n$ *such that* $\phi(\bar{x}) - \phi(x_*) = \Theta_\ell \le \epsilon$*.*

Now, we obtain an upper bound on the number of successful iterations that Algorithm 2.1 takes for finding an ϵ–Pareto critical point. The proof has been adapted from [15], which in turn has been inspired by a technique developed for steepest descent methods in [26].

Theorem 3.2 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 3.1 hold. Then, for any* $\epsilon \in (0,1)$*, Algorithm 2.1 takes at most* $\mathcal{O}(\epsilon^{-1})$ *successful iterations to find a point* $\bar{x}$ *such that* $\omega(\bar{x}) \leq \epsilon$*.*

Proof. Without loss of generality, we can choose $N \geq 1$ such that $|\mathcal{S}_N| \geq 2$. Let $0 \leq \ell < N$ be chosen in a way that $|\mathcal{S}_\ell| \geq 1$, and $|\mathcal{S}_N| \geq 2|\mathcal{S}_\ell|$. In view of Theorem 3.1, we have

$$\phi(x_\ell) - \phi_* \;=\; \Theta_\ell \;\leq\; \frac{1}{|\mathcal{S}_\ell|\eta_1\kappa_2}, \quad \forall \ell \geq 0.$$

Hence, by Assumption 2.4, we have

$$\begin{aligned}
\frac{1}{|\mathcal{S}_\ell|\eta_1\kappa_2} &\geq \phi(x_\ell) - \phi_* \\
&\geq \phi(x_{N+1}) - \phi_* + \phi(x_{\ell+1}) - \phi(x_{N+1}) \\
&= \phi(x_{N+1}) - \phi_* + \sum_{\substack{k=\ell+1 \\ k\in\mathcal{S}}}^{N} \phi(x_k) - \phi(x_{k+1}) \\
&\geq \phi(x_{N+1}) - \phi_* + \eta_1\kappa_{dlam} \sum_{\substack{k=\ell+1 \\ k\in\mathcal{S}}}^{N} \omega(x_k)\min\left\{\frac{\omega(x_k)}{1+\kappa_H}, \Delta_k\right\} \\
&\geq \eta_1\kappa_{dlam}|\mathcal{S}_\ell| \min_{\substack{0\leq k\leq N \\ k\in\mathcal{S}}} \left\{\omega(x_k)\min\left\{\frac{\omega(x_k)}{1+\kappa_H}, \Delta_k\right\}\right\}.
\end{aligned}$$

Thus, there exists $1 \leq k_0 \leq N$ such that

$$\frac{1}{|\mathcal{S}_\ell|^2\eta_1^2\kappa_{dlam}\kappa_2} \;\geq\; \omega(x_{k_0})\min\left\{\frac{\omega(x_{k_0})}{1+\kappa_H}, \Delta_{k_0}\right\}. \tag{21}$$

Now, we consider the two cases of $\frac{\omega(x_{k_0})}{1+\kappa_H} \leq \Delta_{k_0}$ and $\frac{\omega(x_{k_0})}{1+\kappa_H} \geq \Delta_{k_0}$ separately. When $\frac{\omega(x_{k_0})}{1+\kappa_H} \leq \Delta_{k_0}$, in view of (21), we have

$$\frac{1+\kappa_H}{|\mathcal{S}_\ell|^2\eta_1^2\kappa_{dlam}\kappa_2} \;\geq\; \omega(x_{k_0})^2.$$

Thus, for having $\omega(x_{k_0}) \leq \epsilon$, for some $0 \leq k_0 \leq N$, Algorithm 2.1 needs at most

$$|\mathcal{S}_N| = 2|\mathcal{S}_\ell| = 2\left\lceil \frac{\sqrt{1+\kappa_H}}{\kappa_1\eta_1\sqrt{\kappa_{dlam}\kappa_2}}\epsilon^{-1}\right\rceil$$

successful iterations.

On the other hand, when $\frac{\omega(x_{k_0})}{1+\kappa_H} \geq \Delta_{k_0}$, in view of (21), we have

$$\frac{1}{|\mathcal{S}_\ell|^2\eta_1^2\kappa_{dlam}\kappa_2(1+\kappa_H)} \;\geq\; \Delta_{k_0}^2.$$

Thus, in order to obtain $\Delta_{k_0} \leq \kappa_1\epsilon$, for some $0 \leq k_0 \leq N$, Algorithm 2.1 takes at most

$$|\mathcal{S}_N| = 2|\mathcal{S}_\ell| = 2\left\lceil \frac{1}{\kappa_1\eta_1\sqrt{\kappa_{dlam}\kappa_2(1+\kappa_H)}}\epsilon^{-1}\right\rceil$$

successful iterations. Now, in view of Lemma 2.4 with $\kappa = \epsilon$, if $\Delta_{k_0} < \kappa_1\epsilon$ then we have $\omega(x_{k_1}) < \epsilon$ for some $0 \le k_1 \le N$.

Therefore, in both cases the number of successful iterations for driving ω below ϵ is of $\mathcal{O}(\epsilon^{-1})$. ∎

Remark 3.1 *We should mention that, although the technique for the proof of Theorem 3.2 has been adapted from [15], the function ϕ defined at* (5) *is not continuously differentiable unlike the objective function considered in [15].*

Now, by combining Theorem 2.1 and Theorem 3.2, we will have a bound on the total number of iterations the TRMP algorithm requires for deriving an ϵ–Pareto critical point.

Corollary 3.2 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 3.1 hold. Then, for any $\epsilon \in (0, 1)$, Algorithm 2.1 needs at most $\mathcal{O}(\epsilon^{-1})$ iterations for driving ω below ϵ.*

We notice that the bound obtained here matches the one derived for gradient descent multiobjective optimization on Riemannian manifolds [19]. In addition, by setting $m = 1$, we recover the bound derived in [15] for classical trust-region methods for minimization of convex smooth unconstrained single-objective optimization problems. Furthermore, as a byproduct of our WCC analysis, we have derived a WCC bound of $\mathcal{O}(\epsilon^{-1})$ also for trust-region methods for a class of *structured nonsmooth* convex single-objective problems (i.e. pointwise maximum of convex functions).

4 WCC of TRMP algorithm under strong convexity

In this section, we analyze the WCC of TRMP algorithm for minimization of (1) where the component functions are strongly convex. We start by definition of strongly convex function.

Definition 4.1 *[27, Subsection 12.1.2] Function f is called strongly convex with modulus $\mu > 0$ on $\mathbb{R}^n$ if there exists a constant $\mu > 0$ such that for any $x, y \in \mathbb{R}^n$ we have*

$$f(y) \;\ge\; f(x) + (y - x)^\top g_x + \frac{\mu}{2}\|y - x\|^2, \tag{22}$$

for all $g_x \in \partial f(x)$.

The proof presented here is an adaptation of the proof of Theorem 2.1.10 given in [11] for strongly convex smooth functions.

Lemma 4.1 *Let f be a strongly convex function then for any $x, y \in \mathbb{R}^n$, we have*

$$f(y) \;\le\; f(x) + (y - x)^\top g_x + \frac{1}{2\mu}\|g_x - g_y\|^2, \tag{23}$$

for all $g_x \in \partial f(x)$, and $g_y \in \partial f(y)$.

Proof. Let $x \in \mathbb{R}^n$. Let $h(y) = f(y) - y^\top g_x^f$, where $g_x^f \in \partial f(x)$. Since $0 \in \partial h(y) = \partial f(y) - g_x^f$ and h is strongly convex, in view of (22) for any $y \in \mathbb{R}^n$ we have

$$\begin{aligned} h(x) \;&=\; \min_{v \in \mathbb{R}^n} h(v) \;\ge\; \min_{v \in \mathbb{R}^n} \left[h(y) + (v - y)^\top g_y^h + \frac{\mu}{2}\|v - y\|^2 \right] \\ &=\; h(y) - \frac{1}{2\mu}\|g_y^h\|^2, \end{aligned}$$

where $g_y^h \in \partial h(y)$. Thus, we have $h(x) \geq h(y) - \frac{1}{2\mu}\|g_y^h\|^2$, which completes the proof. ■

Assumption 4.1 *For each $i \in I$, the component function f_i is strongly convex with modulus μ_i.*

Under this assumption function ϕ is strongly convex with modulus μ, where $0 < \mu < \min_{i\in I} \mu_i$.

Similarly to the convex case, we establish an upper bound on the distance from the optimal value of ϕ.

Theorem 4.1 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 4.1 hold. Let $\mu < (1+\kappa_H)/2$. Then, we have*

$$\phi(x_\ell) - \phi_* \;\leq\; (1-\eta_1\kappa_3)^{|\mathcal{S}_\ell|}\Theta_0 \;\leq\; \exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0, \tag{24}$$

where

$$\kappa_3 \;=\; 2\kappa_{dlam}\mu \min\left\{\frac{1}{1+\kappa_H}, \kappa_1\right\}.$$

Proof. Due to strong convexity of ϕ, by setting $x = x_*$ and $y = x_k$ in Lemma 4.1, we obtain $\|g_{x_k}\| \geq \sqrt{2\mu\Theta_k}$, for all $g_{x_k} \in \partial\phi(x_k)$, and where Θ_k given in (19). In view of Lemma 3.1, we have $\omega(x_k) = \|g_{0x_k}\|$ for some $g_{0x_k} \in \partial\phi(x_k)$. Hence, we obtain $\omega(x_k) \geq \sqrt{2\mu\Theta_k}$. By applying Lemma 2.4 with $\kappa = \sqrt{2\mu\Theta_k}$, we obtain $\Delta_k \geq \kappa_1\sqrt{2\mu\Theta_k}$. Hence, in view of Assumption 2.4, we have

$$\begin{aligned}
m_k(0) - m(d_k) \;&\geq\; \kappa_{dlam}\omega(x_k)\min\left\{\frac{\omega(x_k)}{1+\kappa_H}, \Delta_k\right\} \\
&\geq\; \kappa_{dlam}\sqrt{2\mu\Theta_k}\min\left\{\frac{\sqrt{2\mu\Theta_k}}{1+\kappa_H}, \kappa_1\sqrt{2\mu\Theta_k}\right\} \\
&\geq\; 2\kappa_{dlam}\mu\Theta_k\min\left\{\frac{1}{1+\kappa_H}, \kappa_1\right\}.
\end{aligned}$$

Thus, for any $k \in \mathcal{S}$, we have

$$\phi(x_k) - \phi(x_{k+1}) \;\geq\; \eta_1\kappa_3\Theta_k. \tag{25}$$

Hence,

$$\Theta_k - \Theta_{k+1} \;\geq\; \eta_1\kappa_3\Theta_k, \quad k \in \mathcal{S}.$$

Therefore,

$$\phi(x_\ell) - \phi_* \;\leq\; (1-\eta_1\kappa_3)^{|\mathcal{S}_\ell|}\Theta_0 \;\leq\; \exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0.$$

■

In view of Theorem 2.1 and Theorem 4.1, the following corollary is readily resulted.

Corollary 4.1 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 4.1 hold. Let $\mu < (1+\kappa_H)/2$. Then, for any $\epsilon \in (0,1)$, Algorithm 2.1 needs at most $\mathcal{O}(\log\epsilon^{-1})$ iterations for finding a point $\bar{x} \in \mathbb{R}^n$ such that $\phi(\bar{x}) - \phi(x_*) = \Theta_\ell \leq \epsilon$.*

In the next theorem we will derive an upper bound on the number of successful iterations for finding an ϵ–Pareto critical point. The proof follows the same machinery as that of Theorem 3.2.

Theorem 4.2 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 4.1 hold. Let $\mu < (1+\kappa_H)/2$. Then, for any $\epsilon > 0$, the total number of successful iterations that Algorithm 2.1 needs to find a point $\bar{x}$ such that $\omega(\bar{x}) \le \epsilon$ is at most $\mathcal{O}(\log \epsilon^{-1})$.*

Proof. Without loss of generality, we can choose $N \ge 1$ such that $|\mathcal{S}_N| \ge 2$. Let $0 \le \ell < N$ be chosen in a way that $|\mathcal{S}_\ell| \ge 1$, and $|\mathcal{S}_N| \ge 2|\mathcal{S}_\ell|$. In view of Theorem 4.1, we have

$$\phi(x_\ell) - \phi_* \ \le \ (1-\eta_1\kappa_3)^{|\mathcal{S}_\ell|}\Theta_0 \ \le \ \exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0.$$

Hence, by Assumption 2.4, we have

$$\begin{aligned}
\exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0 &\ge \phi(x_\ell) - \phi_* \\
&\ge \phi(x_{N+1}) - \phi_* + \phi(x_{\ell+1}) - \phi(x_{N+1}) \\
&= \phi(x_{N+1}) - \phi_* + \sum_{\substack{k=\ell+1\\ k\in\mathcal{S}}}^{N} \phi(x_k) - \phi(x_{k+1}) \\
&\ge \phi(x_{N+1}) - \phi_* + \eta_1\kappa_{dlam} \sum_{\substack{k=\ell+1\\ k\in\mathcal{S}}}^{N} \omega(x_k)\min\left\{\frac{\omega(x_k)}{1+\kappa_H}, \Delta_k\right\} \\
&\ge \eta_1\kappa_{dlam}|\mathcal{S}_\ell| \min_{\substack{0\le k\le N\\ k\in\mathcal{S}}} \left\{\omega(x_k)\min\left\{\frac{\omega(x_k)}{1+\kappa_H}, \Delta_k\right\}\right\}.
\end{aligned}$$

Thus, there exists $1 \le k_0 \le N$ such that

$$\frac{\exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0}{\eta_1\kappa_{dlam}|\mathcal{S}_\ell|} \ \ge \ \omega(x_{k_0})\min\left\{\frac{\omega(x_{k_0})}{1+\kappa_H}, \Delta_{k_0}\right\}.$$

Since, $|\mathcal{S}_\ell| \ge 2$, we have

$$\frac{\exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0}{\eta_1\kappa_{dlam}} \ \ge \ \omega(x_{k_0})\min\left\{\frac{\omega(x_{k_0})}{1+\kappa_H}, \Delta_{k_0}\right\}. \tag{26}$$

Now, we consider two cases:

Case 1: If $\frac{\omega(x_{k_0})}{1+\kappa_H} \le \Delta_{k_0}$, then in view of (26) we have

$$\frac{(1+\kappa_H)\exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0}{\eta_1\kappa_{dlam}} \ \ge \ \omega(x_{k_0})^2.$$

Hence, in order to have $\omega(x_{k_0}) \le \epsilon$, for some $0 \le k_0 \le N$, Algorithm 2.1 takes at most

$$|\mathcal{S}_N| \ = \ 2|\mathcal{S}_\ell| \ = \ \frac{4}{\eta_1\kappa_3}\left\lceil \log\left(\sqrt{\frac{(1+\kappa_H)\Theta_0}{\eta_1\kappa_{dlam}}}\epsilon^{-1}\right)\right\rceil$$

successful iterations.

Case 2: If $\frac{\omega(x_{k_0})}{1+\kappa_H} \ge \Delta_{k_0}$, then, in view of (26), we have

$$\frac{\exp^{-\eta_1\kappa_3|\mathcal{S}_\ell|}\Theta_0}{\eta_1\kappa_{dlam}(1+\kappa_H)} \ \ge \ \Delta_{k_0}^2.$$

Thus, in order to obtain $\Delta_{k_0} \leq \kappa_1 \epsilon$, for some $0 \leq k_0 \leq N$, Algorithm 2.1 takes at most

$$|\mathcal{S}_N| = 2|\mathcal{S}_\ell| = \frac{4}{\eta_1 \kappa_3} \left\lceil \log \left(\sqrt{\frac{\Theta_0}{\eta_1 \kappa_{dlam} \kappa_1^2 (1+\kappa_H)}} \epsilon^{-1} \right) \right\rceil$$

successful iterations. On the other hand, Lemma 2.4 with $\kappa = \epsilon$ implies that if $\Delta_{k_0} \leq \kappa_1 \epsilon$ then $\omega(x_{k_1}) \leq \epsilon$ for some $0 \leq k_1 \leq N$.

Therefore, in both cases the number of successful iterations for driving ω below ϵ is of $\mathcal{O}(\log \epsilon^{-1})$, and the proof is completed. ■

The following corollary, which establishes the complexity of the TRMP algorithm for finding an ϵ–Pareto critical point, is easily resulted from Theorem 2.1 and Theorem 4.1.

Corollary 4.2 *Let Assumptions 2.1, 2.2, 2.3, 2.4, and 4.1 hold. Let* $\mu < (1+\kappa_H)/2$*. Then, for any* $\epsilon > 0$*, the total number of iterations that Algorithm 2.1 takes to drive* ω *below* ϵ *is at most* $\mathcal{O}(\log \epsilon^{-1})$*.*

By setting $m = 1$, we recover the bound derived in [17] for classical trust-region methods for the class of convex smooth unconstrained single-objective optimization problems. Furthermore, as a byproduct of our WCC analysis, we have derived a WCC bound of $\mathcal{O}(\log \epsilon^{-1})$ also for trust-region methods for a class of *structured nonsmooth* strongly convex single-objective problems (i.e. pointwise maximum of strongly convex functions).

5 Conclusions

In this paper, under the presence of (strong) convexity assumption of the component functions, we analyzed the WCC of a multiobjective trust-region method called TRMP, which has originally been presented in [9]. We show that, in order to drive some criticality measure below some predefined positive ϵ, the required number of iterations is at most of $\mathcal{O}(\epsilon^{-1})$. This bound matches the one derived for steepest descent methods for multiobjective optimization methods, and recovers the complexity bound of trust-region in single-objective convex smooth unconstrained optimization. In addition, under the strong convexity of the component functions, we show that the complexity bound is improved to $\mathcal{O}(\log \epsilon^{-1})$, which recovers the complexity bound of trust-region in single-objective strongly convex smooth unconstrained optimization. The techniques used in this work could be easily adapted to derive WCC results for other versions of multiobjective trust-region methods such as the ones developed in [6, 7]. Finally, establishing WCC analysis of the multiobjective version of the NSC framework [23] under the (strong) convexity assumption will be left as future work.

6 Acknowledgments

The author is very grateful to two anonymous referees for very helpful and constructive comments, which significantly improved the contributions and presentation of the paper. This version of the article has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/s11590-022-01932-3. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use
https://www.springernature.com/gp/open-research/policies/acceptedmanuscript-terms

References

[1] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization.* Springer-Verlag Berlin Heidelberg, 2008.

[2] J. Jahn. *Vector optimization.* Springer, 2009.

[3] K. Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

[4] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.*, 51:479–494, 2000.

[5] J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. Newton's method for multiobjective optimization. *SIAM J. Optim.*, 20(2):602–626, 2009.

[6] G. A. Carrizo, P. A. Lotito, and M. C. Maciel. Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem. *Math. Program.*, 159:339–369, 2016.

[7] S. Qu, M. Goh, and B. Liang. Trust region methods for solving multiobjective optimisation. *Optim. Methods Softw.*, 28(4):796–811, 2013.

[8] J. Thomann and G. Eichfelder. A trust-region algorithm for heteregeneous multiobjective optimization. *SIAM J. Optim.*, 29:1017–1047, 2019.

[9] K. D. V. Villacorta, P. R. Oliveira, and A. Soubeyran. A trust-region method for unconstrained multiobjective problems with applications in satisficing processes. *J. Optim. Theory Appl.*, 160(3):865–889, 2014.

[10] E. H. Fukuda and L. M. Graña Drummond. A survey on multiobjective descent methods. *Pesquisa Operacional*, 34(3):585–620, 2014.

[11] Y. Nesterov. *Introductory Lectures on Convex Optimization.* Kluwer Academic Publishers, Dordrecht, 2004.

[12] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. Optim.*, 19:414–444, 2008.

[13] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part II: worst-case function-evaluation complexity. *Math. Program.*, 130:295–319, 2011.

[14] R. Garmanjani, D. Júdice, and L. N. Vicente. Trust-region methods without using derivatives: Worst case complexity and the non-smooth case. *SIAM J. Optim.*, 26:1987–2011, 2016.

[15] R. Garmanjani. A note on the worst-case complexity of nonlinear stepsize control methods for convex smooth unconstrained optimization. *Optimization*, 2020, to appear.

[16] Ph. L. Toint. Nonlinear stepsize control, trust regions and regularizations for unconstrained optimization. *Optim. Methods Softw.*, 28:82–95, 2013.

[17] G.N. Grapiglia, J. Yuan, and Y. Yuan. On the worst-case complexity of nonlinear stepsize control algorithms for convex unconstrained optimization. *Optim. Methods Softw.*, 31:591–604, 2016.

[18] J. Fliege, A. I. F. Vaz, and L. N. Vicente. Complexity of gradient descent for multiobjective optimization. *Optim. Methods Softw.*, 34(5):949–959, 2019.

[19] O. P. Ferreira, M. S. Louzeiro, and L. F. Prudente. Iteration-complexity and asymptotic analysis of steepest decent method for multiobjective optimization on riemannian manifolds. *J. Optim. Theory Appl.*, 184:507–533, 2020.

[20] L. Calderón, M. A. Diniz-Ehrhardt, and J. M. Martínez. On high-order model regularization for multiobjective optimization. *Optim. Methods Softw.*, 2020, to appear.

[21] A. L. Custódio, Y. Diouane, R. Garmanjani, and E. Riccietti. Worst-case complexity bounds of directional direct-search methods for multiobjective derivative-free optimization. *J. Optim. Theory Appl.*, 2020, to appear.

[22] S. Liu and L. N. Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *arXiv preprint arXiv:1907.04472*, 2019.

[23] G.N. Grapiglia, J. Yuan, and Y. Yuan. On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. *Math. Program.*, 152:491–520, 2015.

[24] G. Cocchi and M. Lapucci. An augmented lagrangian algorithm for multi-objective optimization. *Comput. Optim. Appl.*, 77:29–56, 2020.

[25] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.

[26] Y. Nesterov. How to make the gradients small. *Optima*, 88:10–11, 2012.

[27] G.C. Calafiore and L. El Ghaoui. *Optimization Models*. Control systems and optimization series. Cambridge University Press, October 2014.