A decoupled first/second-order steps technique for nonconvex nonlinear unconstrained optimization with improved complexity bounds

S. Gratton* C. W. Royer† L. N. Vicente‡

April 21, 2017

Abstract

In order to be provably convergent towards a second-order stationary point, optimization methods applied to nonconvex problems must necessarily exploit both first and second-order information. However, as revealed by recent complexity analyzes of some of these methods, the overall effort to reach second-order points is significantly larger when compared to the one of approaching first-order ones. In addition, there are other algorithmic schemes, initially designed with first-order convergence in mind, that do not appear to maintain the same first-order performance when modified to take second-order information into account.

In this paper, we propose a technique that separately computes first and second-order steps, and that globally converges to second-order stationary points. Our approach is shown to lead to an improvement of the corresponding complexity bound with respect to the first-order optimality tolerance. Although the applicability of our ideas is wider, we focus the presentation on trust-region methods with and without derivatives.

1 Introduction

1.1 Problem description and motivation

We consider a smooth unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to satisfy the following assumptions.

Assumption 1.1 *The function f is twice continuously differentiable, with Lipschitz continuous gradient and Hessian (and let $L_{\nabla^2 f}$ be the Lipschitz constant of the Hessian).*

*University of Toulouse, IRIT, 2 rue Charles Camichel, B.P. 7122 31071, Toulouse Cedex 7, France (serge.gratton@enseeiht.fr).

†Wisconsin Institute for Discovery, University of Wisconsin-Madison, 330 N Orchard Street, Madison WI 53715, USA (croyer2@wisc.edu). Support for this author was partially provided by Université Toulouse III Paul Sabatier under a doctoral grant.

‡CMUC, Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal (lnv@mat.uc.pt). Support for this author was partially provided by FCT under grants UID/MAT/00324/2013 and P2020 SAICT-PAC/0011/2015.

Assumption 1.2 *The objective function f is bounded below by a value f_{low} on the level-set $\mathcal{L}_f(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$.*

When the objective function is nonconvex, algorithms may exploit both first and second-order information at a given point in order to make progress towards a (local) minimum, at which it is known that the gradient must be zero and the Hessian matrix must be positive semidefinite. Therefore, exploiting directions making an acute angle with a non-zero negative gradient or negative curvature directions corresponding to negative eigenvalues of the Hessian matrix is essential in guaranteeing convergence to such a point. That is also instrumental to the derivation of complexity results, which consists in estimating the worst-case number of iterations (and, often as a by-product, the amount of calls to f and its derivatives) needed to reach an iterate x_k at which

$$\|\nabla f(x_k)\| \ < \ \varepsilon_{\mathtt{C}} \quad \text{and} \quad \left[-\lambda_{\min}(\nabla^2 f(x_k))\right]_+ = \max\left\{-\lambda_{\min}(\nabla^2 f(x_k)), 0\right\} \ < \ \varepsilon_{\mathtt{E}} \tag{1.2}$$

hold, for some given tolerances $\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}} \in (0, 1)$.

As complexity analyses were originally proposed in a convex setting, most of the recently developed results for nonconvex optimization have focused on the worst-case complexity of reaching approximate optimality conditions of only first order (i.e., $\|\nabla f(x_k)\| < \varepsilon_{\mathtt{C}}$). Still, several algorithms were studied from a second-order complexity viewpoint and results related to the satisfaction of (1.2) have been obtained [5, 6, 3, 11, 18, 14, 19] (see also [7] for a generalization to even higher orders)[1].

These algorithms can be classified in two categories. The first one encompasses classes of second-order globally convergent trust-region methods (with derivatives [6] and without [14, 20]), direct-search algorithms [19] for derivative-free optimization, and the general nonlinear stepsize control framework of [18]. In those methods, the complexity bounds for the sole satisfaction of the first criterion in (1.2) are generally of the form $\mathcal{O}(\varepsilon_{\mathtt{C}}^{-2})$. When both criteria in (1.2) are considered, the bound is typically the maximum of two quantities related to the corresponding criteria: $\mathcal{O}\left(\max\left\{\varepsilon_{\mathtt{C}}^{-2}\varepsilon_{\mathtt{E}}^{-1}, \varepsilon_{\mathtt{E}}^{-3}\right\}\right)$ for derivative-based trust-region methods; $\mathcal{O}\left(\max\left\{\varepsilon_{\mathtt{C}}^{-3}, \varepsilon_{\mathtt{E}}^{-3}\right\}\right)$ for direct search. In the remaining cases ([18] and [14, 20]), only the case $\varepsilon_{\mathtt{C}} = \varepsilon_{\mathtt{E}}$ is considered, and a bound of $\mathcal{O}\left(\varepsilon_{\mathtt{E}}^{-3}\right)$ was derived. The presence of a maximum in the above bounds is related to the satisfaction of both conditions in (1.2). In that sense, one may view the first term as characteristic of first-order optimality (related to the gradient norm), and the second as relevant for second-order consideration (tailored to the minimum Hessian eigenvalue). One then observes that the part related to the first-order term worsens in each of those frameworks compared to the case in which only first-order aspects are considered. It indeed becomes $\mathcal{O}(\varepsilon_{\mathtt{C}}^{-2}\varepsilon_{\mathtt{E}}^{-1})$ for trust-region methods and $\mathcal{O}(\varepsilon_{\mathtt{C}}^{-3})$ for direct search.

This phenomenon is not endemic in second-order globally convergent methods. In fact, the second class of algorithms we review here, essentially based upon the cubic regularization analysis derived in [5], does not suffer from this deterioration. Although those frameworks may not explicitly require the minimization of a cubic model [3, 11, 21], they all exhibit complexity bounds of $\mathcal{O}(\varepsilon_{\mathtt{C}}^{-1.5})$ in terms of first-order optimality and $\mathcal{O}(\max\{\varepsilon_{\mathtt{C}}^{-1.5}, \varepsilon_{\mathtt{E}}^{-3}\})$ for second-order optimality. Such results incite to investigate further the reasons for this discrepancy.

[1] We also mention here the recent development of new frameworks inspired by accelerated gradient techniques that provide second-order guarantees while aiming at satisfying approximate first-order optimality [1, 4], despite the fact that these methods cannot be viewed as *full* second-order schemes.

A partial explanation may be found in the theory established for second-order convergent (derivative-based) line-search methods [2, 15, 22, 23]. In such studies, it has been identified that an algorithm exploiting both directions of descent and of negative curvature should not necessarily associate those with the same step length. Indeed, it may be that one of the two criteria of interest (namely the gradient norm or the minimum Hessian eigenvalue) is several orders of magnitude smaller than the other. In that situation, a method based on a unique step length may compute a very small step to cope with the magnitude of one criterion, even though more improvement could have been realized by performing a moderate step if one would focus on the other one.

In trust-region methods, one may also face those issues as any computed step is limited in norm by the trust-region radius. It may be that this radius is forced to shrink in order to provide second-order guarantees. Such aspects are also present in derivative-free optimization, where neither first nor second order derivatives are available. In that setting, the step size (in direct search) or the trust-region radius (in derivative-free trust regions) are often the only available tools to simultaneously estimate both optimality measures. As a result, the cost of the second-order guarantees often overcomes the first-order ones (see [14, 20] and [19]). On the contrary, any technique with a first-order complexity in $\mathcal{O}(\epsilon_{\mathtt{C}}^{-1.5})$ directly relates the norm of the computed step(s) and both criteria of interest in an independent manner.

1.2 Contribution and structure of the paper

Our main motivation is thus the study of algorithms that preserve the original first-order guarantees while additionally taking second-order properties into account, so that the exponent of the first-order power remains unchanged in the overall complexity bound. For this purpose, we present a decoupling technique that dissociates the first and second-order aspects of a given optimization method. Another goal is to address the potential scaling differences that may arise between the optimality measures by using different step sizes.

Our study revolves around the separate treatment of gradient-type and Hessian-type properties of the function at a given iterate. The introduced decoupling technique relies on duplicating elements of the algorithm that intervene in the treatment of both properties, so as to treat each of them separately. In doing so, we will reach a worst-case complexity bound of $\mathcal{O}(\max\{\varepsilon_{\mathtt{C}}^{-2}, \varepsilon_{\mathtt{E}}^{-3}\})$.

As we will see, the idea is general enough to be embedded in a wide range of optimization algorithms. We will however focus on trust-region methods, covering both the derivative-based and the derivative-free cases. The structure of the paper is the following. We present the decoupling concept within a trust-region framework in Section 2. The complexity analysis is derived in Section 3. Some numerical experiments are reported in Section 4 to indicate the potential of our approach. The paper is concluded with a discussion in Section 5.

1.3 Notation

Multiple quantities (models, steps, constants) will be featured in two contexts, respectively related to first and second-order aspects. To avoid confusion, we adopt the superscripts and subscripts $\mathtt{C}$ for first-order quantities, and $\mathtt{E}$ to second-order ones.

The notation $\mathcal{O}(A)$ stands for a scalar times A, with this scalar depending solely on the problem considered or constants from the algorithm. Norms are meant to be Euclidean although we do not directly use this fact.

2 A trust-region method based on decoupled steps

Our presentation of the decoupling technique will be carried on using a basic trust-region paradigm. In the commonly adopted definition of a trust-region method [8, 26], the optimization process consists in the construction a model of the function around the current iterate, followed by a minimization of this model within a trust region (typically defined as a Euclidean ball). If the resulting decrease on the function value is sufficiently large compared to the decrease predicted by the model, the step is accepted and the size of the trust region may be increased. Otherwise, the iterate does not change but the trust-region radius is decreased in an attempt to improve the accuracy of the model by looking at a closer neighborhood [8, 26].

Trust-region methods with second-order guarantees are typically based on the computation of a step which provides a decrease on the model within the trust region comparable to that obtained along the direction of the negative gradient (also called Cauchy decrease) and along the direction of an eigenvector corresponding to the most negative Hessian eigenvalue, if any (also called eigendecrease) [8, Chapter 6]. The optimal solution of the trust-region subproblem also satisfies such a property, and is sometimes considered in second-order global convergence proofs [24, 25].

When one aims to derive complexity results for such methods, it is necessary to relate the function decrease and the step size to the optimality criteria, namely the gradient norm and the minimum Hessian eigenvalue. More specifically, a typical trust-region scheme such as the one analyzed in [6] guarantees that a successful step at iteration k satisfies

$$f(x_{k+1}) - f(x_k) \;\geq\; \mathcal{O}\left(\min\left\{\|\nabla f(x_k)\|\delta_k, \left[-\lambda_{\min}\left(\nabla^2 f(x_k)\right)\right]_+ \delta_k^2\right\}\right), \tag{2.1}$$

where δ_k is the current trust-region radius. It can be additionally established that as long as $\|\nabla f(x_k)\| \geq \varepsilon_{\mathtt{C}}$ or $\lambda_{\min}\left(\nabla^2 f(x_k)\right) \leq -\varepsilon_{\mathtt{E}}$ holds, this radius is bounded away from zero as follows

$$\delta_k \;\geq\; \mathcal{O}\left(\min\{\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}}\}\right). \tag{2.2}$$

This results in an iteration complexity of order $\mathcal{O}(\max\{\varepsilon_{\mathtt{C}}^{-2}\varepsilon_{\mathtt{E}}^{-1}, \varepsilon_{\mathtt{E}}^{-3}\})$. If only the first-order criterion is taken into account, a similar reasoning yields a complexity bound in $\mathcal{O}(\varepsilon_{\mathtt{C}}^{-2})$. The introduction of second-order information appears to worsen the first-order properties of the method, which is not the case for cubic regularization frameworks [5, 6]. This is actually due to the trust-region radius accounting for both optimality criteria, resulting in (2.2). The goal of our decoupling technique is thus to dissociate the two properties in order to recover $\mathcal{O}(\varepsilon_{\mathtt{C}}^{-2})$ in the second-order complexity bound.

2.1 Algorithmic framework

Algorithm 2.1 describes a decoupled version of the traditional second-order globally convergent trust-region method. The critical difference to the classical method is the use of two models,

$$m_k^{\mathtt{C}}(x_k + s) = f_k + (g_k^{\mathtt{C}})^\top s + \frac{1}{2} s^\top H_k^{\mathtt{C}} s, \quad m_k^{\mathtt{E}}(x_k + s) = f_k + (g_k^{\mathtt{E}})^\top s + \frac{1}{2} s^\top H_k^{\mathtt{E}} s,$$

each devoted to capturing information related to the corresponding derivative and approximately minimized within its own trust region. At each iteration, the algorithm thus independently computes two steps that are only connected by a common trust-region parameter δ_k. As a result,

we will see later that each trust-region radius will converge to zero if the method converges to a (true or model) second-order stationary point. In that respect, Algorithm 2.1 follows the same idea than [13] in that it explicitly connects trust-region radii to optimality criteria of interest so as to force the convergence of the trust-region radii sequences. Similar ideas are adopted in the context of derivative-free trust-region methods by the application of the so-called criticality step (see [10]).

After the double step computation, the method chooses the best point with respect to the function value and then computes the maximum of two decrease ratios (where in the numerator one has the actual variation in function value and in denominator the decrease predicted by each step in its model). This significantly differs from the classical approach whose complexity has been analyzed in [6].

2.2 Models

Our first-order model aims to capture gradient information. For this reason, we first require a Taylor-type bound on function values, which will be trivially satisfied in the derivative-based case $g_k^{\mathtt{C}} = \nabla f(x_k)$ with $F_{\mathtt{C}} = (L_{\nabla f} + B_{\mathtt{C}})/2$, where $B_{\mathtt{C}}$ is an upper bound on $H_k^{\mathtt{C}}$.

In the derivative-free case this is achieved by the use of a fully linear model in a ball of radius $\delta_k^{\mathtt{E}}$. The second assumption tightens the accuracy of the model gradient $g_k^{\mathtt{C}}$ to its size. Again, it is trivally satisfied when $g_k^{\mathtt{C}} = \nabla f(x_k)$, this time with $C_{\mathtt{C}} = 0$. In the derivative-free setting this is achieved by means of a criticality step where a fully linear model is computed in a ball of radius proportional to $\|g_k^{\mathtt{C}}\|$ (see [9]).

Assumption 2.1 *For every index k, the corresponding first-order model $m_k^{\mathtt{C}}$ satisfies*

$$\begin{aligned} \left|m_k^{\mathtt{C}}(x_k+s) - f(x_k+s)\right| &\leq F_{\mathtt{C}}(\delta_k^{\mathtt{C}})^2, \quad \forall s \in B(0,\delta_k^{\mathtt{C}}), \\ \left|g_k^{\mathtt{C}} - \nabla f(x_k)\right| &\leq C_{\mathtt{C}}\|g_k^{\mathtt{C}}\|, \end{aligned}$$

where $F_{\mathtt{C}} > 0$ and $C_{\mathtt{C}} \geq 0$ are constants independent of k.

As in the classical case with or without derivatives, we will also need an uniform upper on the model Hessians.

Assumption 2.2 *There exists $B_{\mathtt{C}} > 0$ such that the first-order model Hessian sequence satisfies*

$$\forall k, \ \|H_k^{\mathtt{C}}\| \ \leq \ B_{\mathtt{C}}. \tag{2.5}$$

Similarly, our second-order model aims to capture Hessian information. Now we require the use of a fully quadratic model and tighten the accuracy of the minimum eigenvalue $\lambda_k^{\mathtt{E}}$ of the model Hessian (when negative) to its magnitude.

Assumption 2.3 *For every index k, the corresponding second-order model $m_k^{\mathtt{E}}$ satisfies*

$$\begin{aligned} \left|m_k^{\mathtt{E}}(x_k+s) - f(x_k+s)\right| &\leq F_{\mathtt{E}}(\delta_k^{\mathtt{E}})^3, \quad \forall s \in B(0,\delta_k^{\mathtt{E}}), \\ \left|\left[-\lambda_k^{\mathtt{E}}\right]_+ - \left[-\lambda_k\right]_+\right| &\leq C_{\mathtt{E}}\left[-\lambda_k^{\mathtt{E}}\right]_+ \quad \text{if } \lambda_k^{\mathtt{E}} < 0, \end{aligned}$$

where $F_{\mathtt{E}} > 0$ and $C_{\mathtt{E}} \geq 0$ are positive constants independent of k.

Algorithm 2.1: decoupled Steps in a Trust-REgionS Strategy (DESTRESS)

Choose $x_0 \in \mathbb{R}^n$, $0 < \delta_0 < \delta_{\max}$, $0 < \gamma_1 < 1 \leq \gamma_2$, and $\eta > 0$.
for $k = 0, 1, 2, \ldots$ **do**

1. **First-order trust-region step**

   (a) Compute a model $m_k^{\mathtt{C}}$ of the function f, and a step $s_k^{\mathtt{C}}$ that approximately solves the first-order trust-region subproblem

   $$\begin{cases} \min_s \; m_k^{\mathtt{C}}(x_k + s) \\ \|s\| \leq \delta_k^{\mathtt{C}} \stackrel{\text{def}}{=} \delta_k \left\|g_k^{\mathtt{C}}\right\|. \end{cases} \tag{2.3}$$

   (b) Set $x_k^{\mathtt{C}} = x_k + s_k^{\mathtt{C}}$ and compute $f(x_k^{\mathtt{C}})$.

2. **Second-order trust-region step**

   (a) Compute a model $m_k^{\mathtt{E}}$ of the function f. If $\lambda_k^{\mathtt{E}} < 0$, compute a step $s_k^{\mathtt{E}}$ that approximately solves the second-order trust-region subproblem

   $$\begin{cases} \min_s \; m_k^{\mathtt{E}}(x_k + s) \\ \|s\| \leq \delta_k^{\mathtt{E}} \stackrel{\text{def}}{=} \delta_k \left[-\lambda_k^{\mathtt{E}}\right]_+. \end{cases} \tag{2.4}$$

   If $\lambda_k^{\mathtt{E}} \geq 0$, no second-order step is computed.

   (b) Set $x_k^{\mathtt{E}} = x_k + s_k^{\mathtt{E}}$ and compute $f(x_k^{\mathtt{E}})$.

3. **Decrease ratio and iterate update**

   (a) If a second-order step was computed, choose $s_k \in \arg\min \left\{f(x_k + s_k^{\mathtt{C}}), f(x_k + s_k^{\mathtt{E}})\right\}$ and set

   $$\rho_k \;=\; \max\left\{\rho_k^{\mathtt{C}}, \rho_k^{\mathtt{E}}\right\}$$

   with

   $$\rho_k^{\mathtt{C}} \;=\; \frac{f(x_k) - f(x_k + s_k)}{m_k^{\mathtt{C}}(x_k) - m_k^{\mathtt{C}}(x_k + s_k^{\mathtt{C}})} \quad \text{and} \quad \rho_k^{\mathtt{E}} \;=\; \frac{f(x_k) - f(x_k + s_k)}{m_k^{\mathtt{E}}(x_k) - m_k^{\mathtt{E}}(x_k + s_k^{\mathtt{E}})}.$$

   Otherwise, $s_k = s_k^{\mathtt{C}}$ and $\rho_k = \rho_k^{\mathtt{C}}$.

   (b) If $\rho_k \geq \eta$, set $x_{k+1} = x_k + s_k$ and declare the iteration as successful, otherwise declare the iteration as unsuccessful.

4. **Trust-region parameter update**

   Set

   $$\delta_{k+1} \;=\; \begin{cases} \min\left\{\gamma_2 \delta_k, \delta_{\max}\right\} & \text{if } \rho_k \geq \eta, \\ \gamma_1 \delta_k & \text{otherwise.} \end{cases}$$

When using a Taylor model ($g_k^{\mathtt{E}} = \nabla f(x_k)$, $H_k^{\mathtt{E}} = \nabla^2 f(x_k)$), one trivially has $F_{\mathtt{E}} = L_{\nabla^2 f}/6$ and $C_{\mathtt{E}} = 0$.

In the derivative-free case the first assumption is achieved by the use of a fully quadratic model in a ball of radius $\delta_k^{\mathtt{E}}$ and the second one by a criticality step where a fully quadratic model is computed in a ball of radius proportional to $\left[-\lambda_k^{\mathtt{E}}\right]_+$ when $\lambda_k^{\mathtt{E}}$ is negative (as in [9]). We must point out here that the strict satisfaction of the second assumption in a derivative-free context would be hard to do in a finite number of iterations within a criticality-type step. However, we observe that in order to still garantee the result (Lemma 3.2) where such assumption is used, one can relax it as

$$\left|\left[-\lambda_k^{\mathtt{E}}\right]_+ - \left[-\lambda_k\right]_+\right| \;\leq\; C_{\mathtt{E}}\left[-\lambda_k^{\mathtt{E}}\right]_+ + \frac{\varepsilon_{\mathtt{E}}}{2} \quad \text{if} \quad \lambda_k^{\mathtt{E}} \;\leq\; \varepsilon_{\mathtt{E}},$$

where $\varepsilon_{\mathtt{E}} > 0$ is the second-order tolerance for the derivation of the complexity bound.

In the derivative-based case, where again the models are quadratic functions based on the first and second-order Taylor expansions of f around the current iterate, an iteration of the DESTRESS algorithm requires one gradient and one Hessian evaluation, together with two calls to the objective function.

2.3 Subproblem solution

On the first-order side, we impose on the approximate subproblem solution the classical requirement of first-order convergent trust-region methods.

Assumption 2.4 *At each iteration k of Algorithm 2.1, the approximate solution of the trust-region subproblem (2.3) satisfies a fraction of Cauchy decrease, i.e., the first-order step $s_k^{\mathtt{C}}$ satisfies*

$$m_k^{\mathtt{C}}(x_k) - m_k^{\mathtt{C}}(x_k + s_k^{\mathtt{C}}) \;\geq\; \tau_{\mathtt{C}} \|g_k^{\mathtt{C}}\| \min\left\{\frac{\|g_k^{\mathtt{C}}\|}{\|H_k^{\mathtt{C}}\|}, \delta_k^{\mathtt{C}}\right\}, \tag{2.6}$$

where $\tau_{\mathtt{C}} \in (0, \frac{1}{2}]$ and we set $\|g_k^{\mathtt{C}}\|/\|H_k^{\mathtt{C}}\| = \infty$ whenever $\|H_k^{\mathtt{C}}\| = 0$.

Note that with our specific definition of the first-order trust-region radius, (2.6) reduces to

$$m_k^{\mathtt{C}}(x_k) - m_k^{\mathtt{C}}(x_k + s_k^{\mathtt{C}}) \;\geq\; \tau_{\mathtt{C}} \|g_k^{\mathtt{C}}\|^2 \min\left\{\frac{1}{\left\|H_k^{\mathtt{C}}\right\|}, \delta_k\right\}. \tag{2.7}$$

A similar requirement is imposed on the second-order side.

Assumption 2.5 *At each iteration k of Algorithm 2.1, the approximate solution of the trust-region subproblem (2.4) satisfies a fraction of eigendecrease, i.e., when $\left[-\lambda_k^{\mathtt{E}}\right]_+ > 0$ the second-order step $s_k^{\mathtt{E}}$ satisfies*

$$m_k^{\mathtt{E}}(x_k) - m_k^{\mathtt{E}}(x_k + s_k^{\mathtt{E}}) \;\geq\; \tau_{\mathtt{E}}\left[-\lambda_k^{\mathtt{E}}\right]_+ \left[-\delta_k^{\mathtt{E}}\right]^2, \tag{2.8}$$

where $\tau_{\mathtt{E}} \in (0, 1]$.

As before, we observe that our definition of $\delta_k^{\mathtt{E}}$ yields

$$m_k^{\mathtt{E}}(x_k) - m_k^{\mathtt{E}}(x_k + s_k^{\mathtt{E}}) \;\geq\; \tau_{\mathtt{E}}\left[-\lambda_k^{\mathtt{E}}\right]_+^3 \delta_k^2. \tag{2.9}$$

Assumptions 2.4 and 2.5 are typically satisfied, respectively, by a step along the direction of the negative gradient (also called Cauchy step) and a step along the direction of an eigenvector associated to $\lambda_k^{\mathtt{E}} > 0$ (also called eigenstep).

2.4 Basic results for step acceptance

The model properties we enforced are instrumental to guarantee progress towards a solution of problem (1.1). Indeed, when the models are chosen to be sufficiently accurate approximations of the objective function and the trust-region radii are sufficiently small, steps associated with the subproblems will produce satisfying decrease in the function value and be accepted as new iterates. This is the sense of the following lemmas.

Lemma 2.1 *Let Assumptions 1.1, 2.1, 2.2, and 2.4 hold. If* $\|g_k^{\mathtt{C}}\| > 0$ *and*

$$\delta_k^{\mathtt{C}} \;<\; \min\left\{\frac{1}{B_{\mathtt{C}}}, \frac{\tau_{\mathtt{C}}(1-\eta)}{F_{\mathtt{C}}}\right\}\|g_k^{\mathtt{C}}\|, \tag{2.10}$$

then the k-th iteration is first-order successful and the first-order trust-region parameter is not decreased.

Proof. Since we have $\rho_k \geq \rho_k^{\mathtt{C}}$, it suffices to prove that $\rho_k^{\mathtt{C}} \geq \eta$. One has:

$$\begin{aligned}
|\rho_k^{\mathtt{C}} - 1| &= \left|\frac{f(x_k) - f(x_k + s_k^{\mathtt{C}}) - m_k^{\mathtt{C}}(x_k) + m_k^{\mathtt{C}}(x_k + s_k^{\mathtt{C}})}{m_k^{\mathtt{C}}(x_k) - m_k^{\mathtt{C}}(x_k + s_k^{\mathtt{C}})}\right| \\
&= \frac{\left|m_k^{\mathtt{C}}(x_k + s_k^{\mathtt{C}}) - f(x_k + s_k^{\mathtt{C}})\right|}{\left|m_k^{\mathtt{C}}(x_k) - m_k^{\mathtt{C}}(x_k + s_k)\right|} \\
&\leq \frac{F_{\mathtt{C}}\left[\delta_k^{\mathtt{C}}\right]^2}{\tau_{\mathtt{C}}\|g_k^{\mathtt{C}}\| \min\left\{\frac{\|g_k^{\mathtt{C}}\|}{\|H_k^{\mathtt{C}}\|}, \delta_k^{\mathtt{C}}\right\}} \\
&\leq \frac{F_{\mathtt{C}}\left[\delta_k^{\mathtt{C}}\right]^2}{\tau_{\mathtt{C}}\|g_k^{\mathtt{C}}\| \min\left\{\frac{\|g_k^{\mathtt{C}}\|}{B_{\mathtt{C}}}, \delta_k^{\mathtt{C}}\right\}} \\
&\leq \frac{F_{\mathtt{C}}\delta_k^{\mathtt{C}}}{\tau_{\mathtt{C}}\|g_k^{\mathtt{C}}\|} \\
&\leq 1 - \eta,
\end{aligned}$$

where the last two inequalities are direct consequences of (2.10). □

Lemma 2.2 *Let Assumptions 1.1, 2.3, and 2.5 hold. Suppose that at iteration k,* $\lambda_k^{\mathtt{E}} < 0$ *and Step 2 of Algorithm 2.1 is reached with*

$$\delta_k^{\mathtt{E}} \;\leq\; \frac{\tau_{\mathtt{E}}(1-\eta)}{F_{\mathtt{E}}}\left[-\lambda_k^{\mathtt{E}}\right]_+. \tag{2.11}$$

Then, the k-th iteration is second-order successful and the second-order trust-region parameter is not decreased.

Proof. Similarly to the proof of Lemma 2.1, one then has:

$$\begin{aligned} |\rho_k^{\mathtt{E}} - 1| &= \left| \frac{f(x_k) - f(x_k + s_k^{\mathtt{E}}) - m_k^{\mathtt{E}}(x_k) + m_k^{\mathtt{E}}(x_k + s_k^{\mathtt{E}})}{m_k^{\mathtt{E}}(x_k) - m_k^{\mathtt{E}}(x_k + s_k^{\mathtt{E}})} \right| \\ &= \frac{\left| m_k^{\mathtt{E}}(x_k + s_k^{\mathtt{E}}) - f(x_k + s_k^{\mathtt{E}}) \right|}{\left| m_k^{\mathtt{E}}(x_k) - m_k^{\mathtt{E}}(x_k + s_k) \right|} \\ &\leq \frac{F_{\mathtt{E}} \left[\delta_k^{\mathtt{E}}\right]^3}{\tau_{\mathtt{E}}[-\lambda_k^{\mathtt{E}}]_+ \left[\delta_k^{\mathtt{E}}\right]^2} \\ &\leq \frac{F_{\mathtt{E}} \left[\delta_k^{\mathtt{E}}\right]}{\tau_{\mathtt{E}}[-\lambda_k^{\mathtt{E}}]_+} \\ &\leq 1 - \eta, \end{aligned}$$

hence $\rho_k \geq \rho_k^{\mathtt{E}} \geq \eta$. □

3 Worst case complexity

As an auxiliary result, we can exploit the results of Lemmas 2.1 and 2.2 to provide lower bounds on the trust-region parameter at points sufficiently away from second-order stationarity (or model stationarity in the derivative-free case).

Lemma 3.1 *Let Assumptions 1.1, 2.1, 2.3, 2.4, and 2.5 hold. Suppose that by the k-th iteration, the method has not reached a (true or model) second-order stationary point, meaning that $\forall l \leq k$ either $\|g_l^{\mathtt{C}}\| > 0$ or $[-\lambda_l^{\mathtt{E}}]_+ > 0$.*

Then, for every $l \leq k$, one has

$$\delta_l \;\geq\; \frac{\gamma_1}{\gamma_2} \min\left\{ \frac{1}{B_{\mathtt{C}}}, \frac{\tau_{\mathtt{C}}(1-\eta)}{F_{\mathtt{C}}}, \frac{\tau_{\mathtt{E}}(1-\eta)}{F_{\mathtt{E}}} \right\}. \tag{3.1}$$

Proof. For the purpose of deriving a contradiction, suppose that l is the first iterate such that

$$\delta_{l+1} \;<\; \gamma_1 \min\left\{ \frac{1}{B_{\mathtt{C}}}, \frac{\tau_{\mathtt{C}}(1-\eta)}{F_{\mathtt{C}}}, \frac{\tau_{\mathtt{E}}(1-\eta)}{F_{\mathtt{E}}} \right\}. \tag{3.2}$$

By the updating rules on the trust-region parameter, we have that $\delta_{l+1} \geq \gamma_1 \delta_l$, so

$$\delta_l \;<\; \min\left\{ \frac{1}{B_{\mathtt{C}}}, \frac{\tau_{\mathtt{C}}(1-\eta)}{F_{\mathtt{C}}}, \frac{\tau_{\mathtt{E}}(1-\eta)}{F_{\mathtt{E}}} \right\}$$

also holds. Given that we have not reached a model second-order stationary point by assumption, we know that either $\|g_l^{\mathtt{C}}\| > 0$ or $[-\lambda_l^{\mathtt{E}}]_+ > 0$.

Suppose first that $\|g_l^{\mathtt{C}}\| > 0$. Then one has

$$\delta_l^{\mathtt{C}} \;=\; \delta_l \|g_l^{\mathtt{C}}\| \;\leq\; \min\left\{ \frac{1}{B_{\mathtt{C}}}, \frac{\tau_{\mathtt{C}}(1-\eta)}{L_{\mathtt{C}}} \right\} \|g_l^{\mathtt{C}}\|. \tag{3.3}$$

We thus have from Lemma 2.1 that iteration l is successful and the trust-region parameter is not decreased. As a result, $\delta_{l+1} \geq \delta_l$, which contradicts the assumption that l is the first iteration index satisfying (3.2).

Suppose now that $[-\lambda_l^{\mathtt{E}}]_+ > 0$. In that case,

$$\delta_l^{\mathtt{E}} \;\leq\; \frac{\tau_{\mathtt{E}}(1-\eta)}{L_{\mathtt{E}}}\left[-\lambda_l^{\mathtt{E}}\right]_+ \tag{3.4}$$

and by Lemma 2.2, iteration l is successful without decreasing the trust-region parameter. One then draws the same contradiction than in the first case. From both cases we conclude that (3.2) cannot hold, and thus for every $l \leq k$,

$$\gamma_1 \min\left\{\frac{1}{B_{\mathtt{C}}}, \frac{\tau_{\mathtt{C}}(1-\eta)}{F_{\mathtt{C}}}, \frac{\tau_{\mathtt{E}}(1-\eta)}{F_{\mathtt{E}}}\right\} \;\leq\; \delta_{l+1} \;\leq \gamma_2 \delta_l,$$

hence the result. □

Our goal is now to bound the number of iterations that Algorithm 2.1 needs to reach an $(\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}})$-*approximate second-order stationary point*, that is a point at which both

$$\|\nabla f(x_k)\| < \varepsilon_{\mathtt{C}} \tag{3.5}$$

and

$$[\lambda_k]_+ < \varepsilon_{\mathtt{E}}, \tag{3.6}$$

hold, with $(\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}}) \in (0,1)^2$. To establish such a worst-case complexity bound, we define k_ε as the first integer such that both (3.5) and (3.6) hold. Besides, we let S_ε and U_ε denote the set of successful iterations and unsuccessful iterations, i.e., the set of iterations of index less than or equal to k_ε for which $\rho_k \geq \eta$ and $\rho_k < \eta$, respectively.

Lemma 3.2 *Let Assumptions 1.1, 1.2, 2.1, 2.3, 2.4, and 2.5 hold. Then,*

$$|S_\varepsilon| \;\leq\; \frac{f(x_0) - f_{\text{low}}}{\mathcal{C}} \max\left\{\varepsilon_{\mathtt{C}}^{-2}, \varepsilon_{\mathtt{E}}^{-3}\right\}, \tag{3.7}$$

where

$$\mathcal{C} \;=\; \eta \min\left\{\frac{\tau_{\mathtt{C}}\kappa_\delta}{(1+C_{\mathtt{C}})^2}, \frac{\tau_{\mathtt{E}}\kappa_\delta^2}{(1+C_{\mathtt{E}})^3}\right\}, \quad \kappa_\delta \;=\; \frac{\gamma_1}{\gamma_2}\min\left\{\frac{1}{B_{\mathtt{C}}}, \frac{\tau_{\mathtt{C}}(1-\eta)}{F_{\mathtt{C}}}, \frac{\tau_{\mathtt{E}}(1-\eta)}{F_{\mathtt{E}}}\right\}. \tag{3.8}$$

Proof. Let $k \leq k_\varepsilon$ be the index of a successful iteration. Then, either (3.5) or (3.6) does not hold.

In the first case,

$$-\left\|g_k^{\mathtt{C}} - \nabla f(x_k)\right\| + \|\nabla f(x_k)\| \;\leq\; \left\|g_k^{\mathtt{C}}\right\|,$$

which by Assumption 2.2 implies then

$$\left\|g_k^{\mathtt{C}}\right\| \;\geq\; \frac{\varepsilon_{\mathtt{C}}}{1+C_{\mathtt{C}}}.$$

Hence, using (2.7),

$$m_k^{\mathtt{C}}(x_k) - m_k^{\mathtt{C}}(x_k + s_k^{\mathtt{C}}) \;\geq\; \tau_{\mathtt{C}}\|g_k^{\mathtt{C}}\|^2 \min\left\{\frac{1}{B_{\mathtt{C}}}, \delta_k\right\} \;\geq\; \frac{\tau_{\mathtt{C}}\varepsilon_{\mathtt{C}}^2\kappa_\delta}{(1+C_{\mathtt{C}})^2},$$

where we also applied the result of Lemma 3.1 together with the fact that $\kappa_\delta < 1/B_{\mathtt{C}}$.

In the second case, we know from Assumption 2.2 that $\lambda_k^{\mathtt{E}} < 0$. Hence, we obtain

$$-\left|\left[-\lambda_k^{\mathtt{E}}\right]_+ - \left[-\lambda_k\right]_+\right| + \left[-\lambda_k\right]_+ \;\leq\; \left[-\lambda_k^{\mathtt{E}}\right]_+,$$

which by Assumption 2.2 implies then

$$\left[-\lambda_k^{\mathtt{E}}\right]_+ \;\geq\; \frac{\varepsilon_{\mathtt{E}}}{1+C_{\mathtt{E}}}.$$

As a result,

$$m_k^{\mathtt{E}}(x_k) - m_k^{\mathtt{E}}(x_k + s_k^{\mathtt{E}}) \;\geq\; \tau_{\mathtt{E}}\left[-\lambda_k\right]_+^3 \delta_k^2 \;\geq\; \frac{\tau_{\mathtt{E}}\varepsilon_{\mathtt{E}}^3\kappa_\delta^2}{(1+C_{\mathtt{E}})^3}.$$

Putting the two bounds together, we obtain that the function decrease at the k-th iteration satisfies

$$f(x_k) - f(x_{k+1}) \;\geq\; \eta \min\left\{\frac{\tau_{\mathtt{C}}\varepsilon_{\mathtt{C}}^2\kappa_\delta}{(1+C_{\mathtt{E}})^2}, \frac{\tau_{\mathtt{E}}\varepsilon_{\mathtt{E}}^3\kappa_\delta^2}{(1+C_{\mathtt{E}})^3}\right\}. \tag{3.9}$$

By considering the sum of the decreases across all iterations and using Assumption 1.2, we obtain:

$$\begin{aligned} f(x_0) - f_{\text{low}} &\geq \sum_{k\leq k_\varepsilon} f(x_k) - f(x_{k+1}) \\ &= \sum_{k\in S_\varepsilon} f(x_k) - f(x_{k+1}) \\ &\geq |S_\varepsilon|\,\eta \min\left\{\frac{\tau_{\mathtt{C}}\varepsilon_{\mathtt{C}}^2\kappa_\delta}{(1+C_{\mathtt{E}})^2}, \frac{\tau_{\mathtt{E}}\varepsilon_{\mathtt{E}}^3\kappa_\delta^2}{(1+C_{\mathtt{E}})^3}\right\} \\ &\geq |S_\varepsilon|\,\mathcal{C}\min\left\{\varepsilon_{\mathtt{C}}^2, \varepsilon_{\mathtt{E}}^3\right\}, \end{aligned}$$

hence the result. □

Lemma 3.3 *Under the assumptions of Lemma 3.2, one has*

$$|U_\varepsilon| \;\leq\; \log_{\gamma_1}\left(\delta_0^{-1}\kappa_\delta\right) - \log_{\gamma_1}(\gamma_2)|S_\varepsilon|. \tag{3.10}$$

Proof. From the update formulas on δ_k, one has

$$\delta_{k_\varepsilon} \;\leq\; \delta_0\gamma_1^{|U_\varepsilon|}\gamma_2^{|S_\varepsilon|}.$$

Taking logarithms, one obtains

$$-\log(\gamma_1)|U_\varepsilon| \;\leq\; \log(\delta_0) - \log(\delta_{k_\varepsilon}) + \log(\gamma_2)|S_\varepsilon|.$$

After division by $-\log(\gamma_1) > 0$, this becomes:

$$|U_\varepsilon| \;\leq\; -\log_{\gamma_1}(\delta_0) + \log_{\gamma_1}(\delta_{k_\varepsilon}) - \log_{\gamma_1}(\gamma_2)|S_\varepsilon|.$$

Since δ_{k_ε} satisfies (3.1) and κ_δ is given by (3.8), we obtain the desired result □

We finally obtain our complexity bound by applying $k_\varepsilon = |S_\varepsilon| + |U_\varepsilon|$. The result is given below in Theorem 3.1.

Theorem 3.1 *Let Assumptions of Lemma 3.2 hold. Then, the number of iterations needed to attain an* $(\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}})$*-approximate second-order stationary point is*

$$\mathcal{O}\left(\max\left\{\varepsilon_{\mathtt{C}}^{-2}, \varepsilon_{\mathtt{E}}^{-3}\right\}\right), \tag{3.11}$$

where the constant in $\mathcal{O}(\cdot)$ *does not depend on* $\varepsilon_{\mathtt{C}}$ *or* $\varepsilon_{\mathtt{E}}$*, but on* f_{low}*,* $f(x_0)$*,* $F_{\mathtt{C}}$*,* $F_{\mathtt{E}}$*,* $C_{\mathtt{C}}$*,* $C_{\mathtt{E}}$*,* $\tau_{\mathtt{C}}$*,* $\tau_{\mathtt{E}}$*,* γ_1*,* γ_2*,* η*, and* δ_0*.*

To end this section, we point out that Theorem 3.1 implies a liminf-type global convergence result of the form

$$\liminf_{k\to\infty} \max\left\{\|\nabla f(x_k)\|, -\lambda_{\min}(\nabla^2 f(x_k))\right\}.$$

4 A numerical illustration

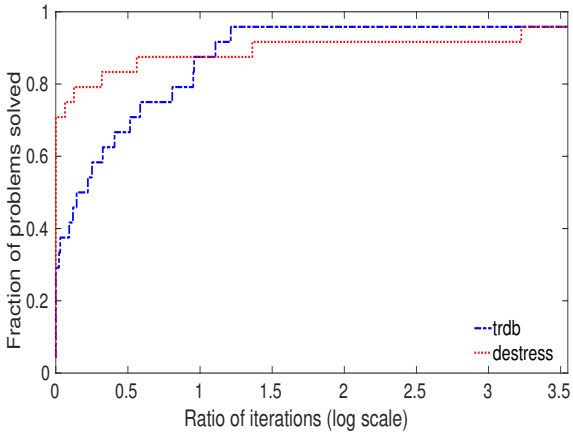
In order to illustrate the effect of our approach, we selected a benchmark of 58 nonconvex problems from the CUTEst collection [16] for which negative curvature was detected around the initial point (see [2]). We implemented a standard trust-region method (denoted by `trdb`) and Algorithm 2.1 in MATLAB. For both methods, we used exact second-order Taylor models, and we computed Cauchy steps and steps along eigenvectors corresponding to the most negative eigenvalue (if any): as pointed above, those steps satisfy the necessary requirements for our complexity analysis. We set the initial trust-region parameter to $\delta_0 = 1$ (note that in the case of the classical trust-region framework, this represents the value of the trust-region radius). In addition, we set $\gamma_1 = \gamma_2^{-1} = 0.5$, $\eta = 0.25$, and $\delta_{\max} = \infty$.

We build performance profiles [12] using the number of iterations as performance metric. Note that it also corresponds to the number of gradient and Hessian evaluations. In order to enlighten the specificities of our method, we adopt the standard approach of removing from the profiles the problems for which both methods had the same performance.

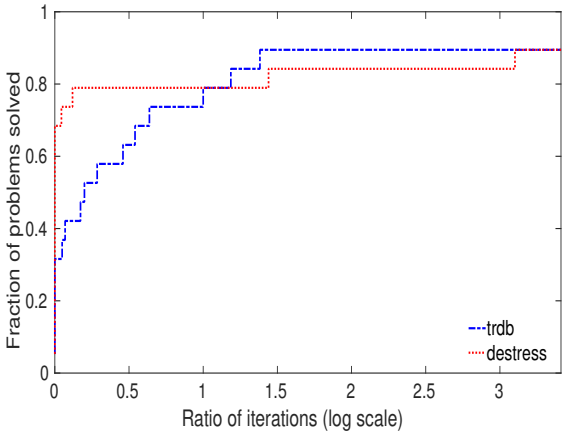
During our experiments, we found out that both methods quickly reached a region where the second-order criterion was satisfied for all iterates (the Hessian had no negative eigenvalues or a slightly nonpositive one): the variation in our profiles was thus essentially caused by a change upon the tolerance on the norm of the gradient. Therefore, we will restrict the presentation of the results to a single choice of the second-order tolerance, namely $\varepsilon_{\mathtt{E}} = 10^{-3}$.

Figure 1 and 2 correspond to the profiles obtained by considering the tolerances $(\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}})$ as used in our convergence analysis. One observes that the `destress` algorithm is generally more efficient than the `trdb` method in that it requires less iterations to reach an approximate stationary point (therefore the `destress` curve lies above the `trdb` curve on the y-axis). However, the classical trust-region approach is the highest curve as the ratio gets larger, indicating that it is able to solve more problems within the given budget. This trend is more noticeable using a large budget of 10000 iterations (Figure 2); when the budget is relatively moderate (500 iterations, Figure 1), the two curves eventually coincide for large values of the iteration ratio.

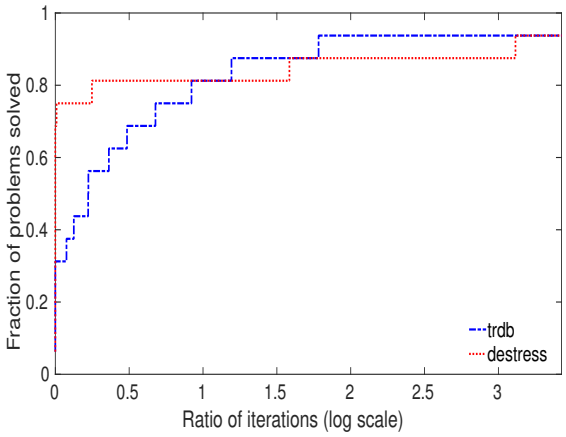
From our results, we can infer that the decoupled approach takes advantage of the iterations for which one optimality measure has a large value. In that situation, a large step can be taken within the appropriate trust-region, and this step is guaranteed to yield a decrease if the trust-region *parameter* (as opposed to the trust-region *radius*) is small. However, when high accuracy is required, the situation might be the opposite in that the trust-region *radii* of the decoupled strategy would then inevitably shrink, restricting the steps to be small. Although the resulting



(a) $\varepsilon_{\mathsf{c}} = 10^{-3}$.

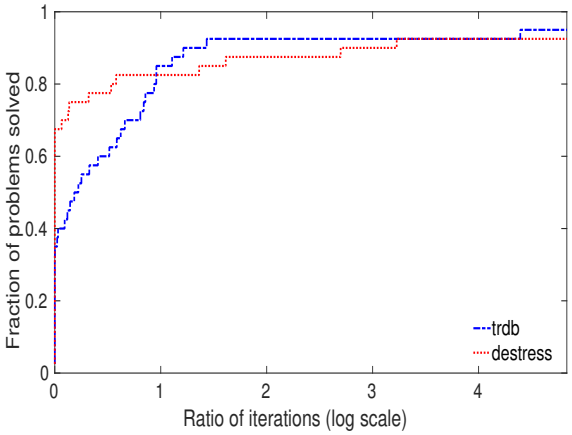


(b) $\varepsilon_{\mathsf{c}} = 10^{-4}$.

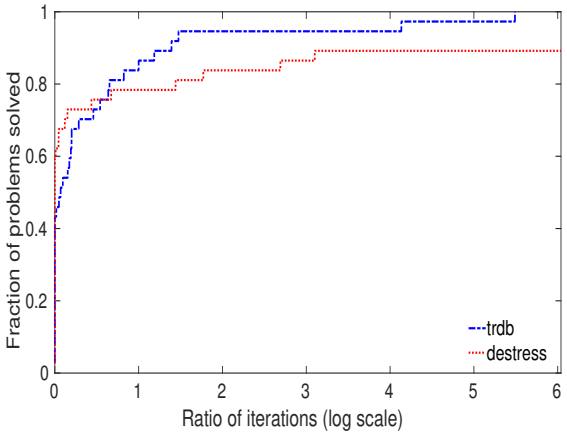


(c) $\varepsilon_{\mathsf{c}} = 10^{-6}$.

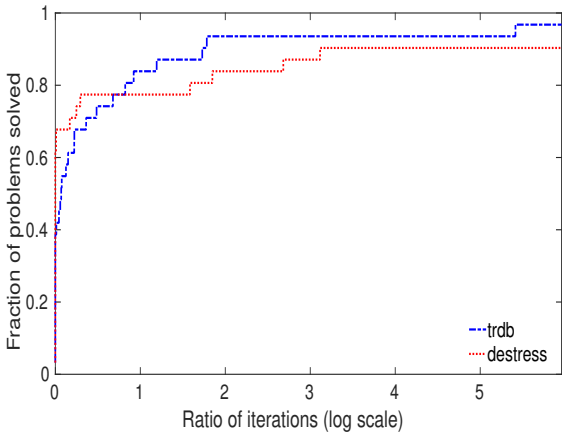
Figure 1: Performance of standard and decoupled trust-region methods given a budget of 500 iterations.



(a) $\varepsilon_{\mathsf{c}} = 10^{-3}$.

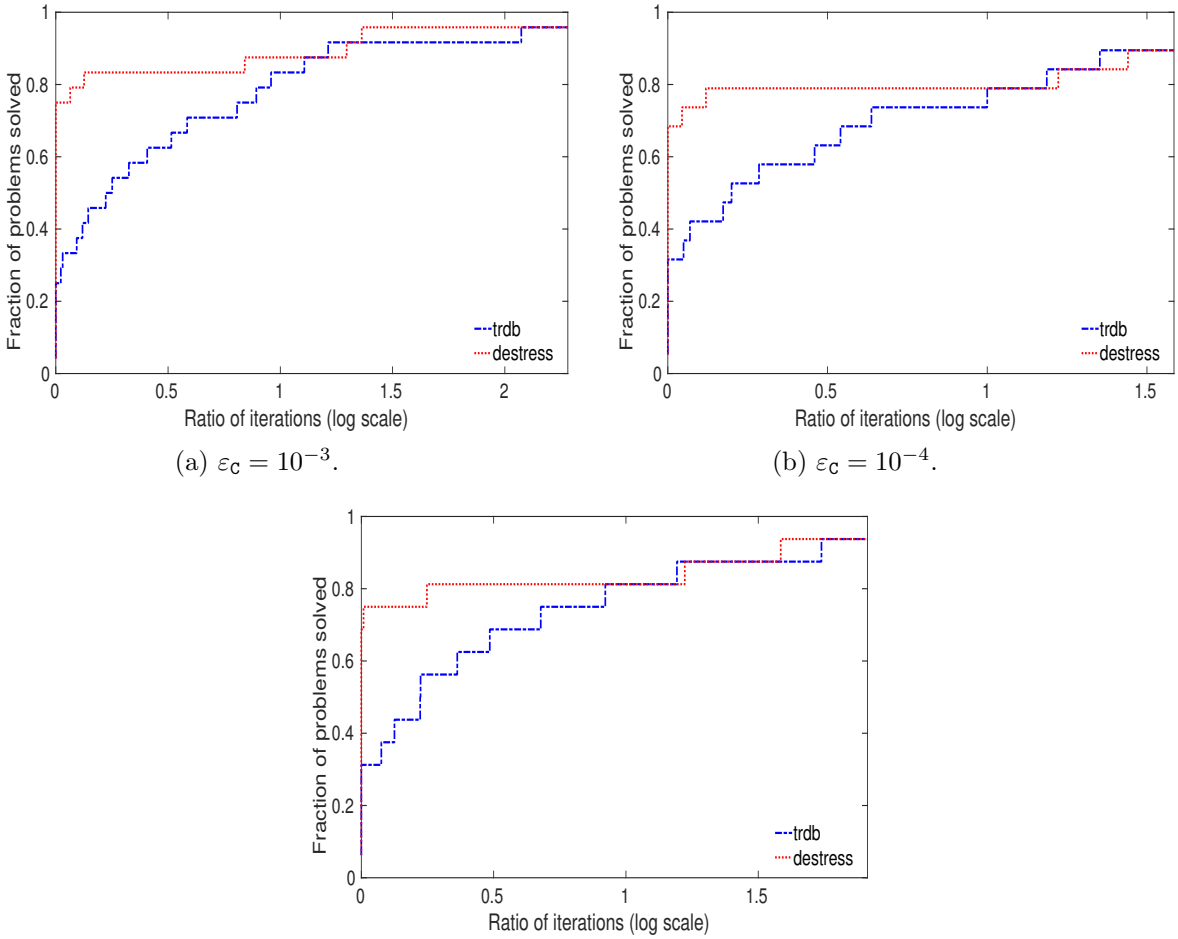


(b) $\varepsilon_{\mathsf{c}} = 10^{-4}$.



(c) $\varepsilon_{\mathsf{c}} = 10^{-6}$.

Figure 2: Performance of standard and decoupled trust-region methods given a budget of 10000 iterations.



(a) $\varepsilon_{\mathtt{C}} = 10^{-3}$.

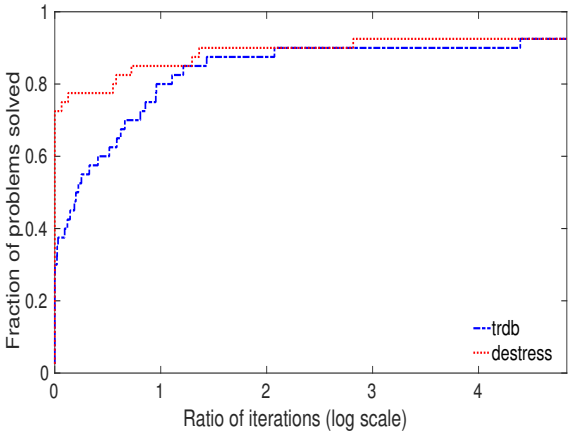
(b) $\varepsilon_{\mathtt{C}} = 10^{-4}$.

(c) $\varepsilon_{\mathtt{C}} = 10^{-6}$.

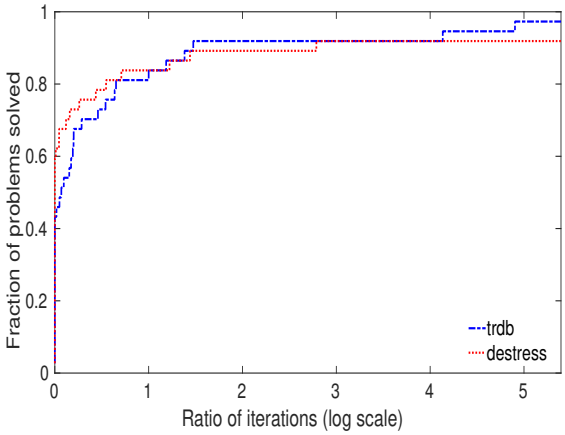
Figure 3: Performance of standard and decoupled (with alternative rule) trust-region methods given a budget of 500 iterations.

decrease would be acceptable from a complexity point of view, it could limit the progress made by the method on a single iteration.

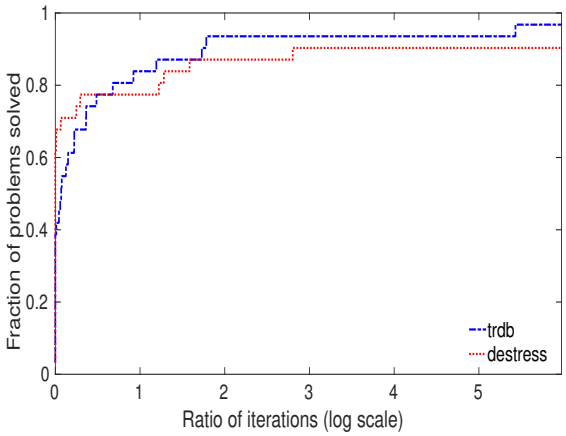
To confirm this idea, we modified the `destress` method so that the trust-region radii are given by $\delta_k^{\mathtt{C}} = \delta_k \max\left\{\|g_k^{\mathtt{C}}\|, \varepsilon_{\mathtt{C}}\right\}$ and $\delta_k^{\mathtt{E}} = \delta_k \max\left\{[-\lambda_k^{\mathtt{E}}]_+, \varepsilon_{\mathtt{E}}\right\}$. Note that our complexity analysis using the tolerances $(\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}})$ is not affected by these changes, since it relies on iterations for which either $\|g_k^{\mathtt{C}}\| \geq \varepsilon_{\mathtt{C}}$ or $[-\lambda_k^{\mathtt{E}}]_+ \geq \varepsilon_{\mathtt{E}}$, in which case the corresponding trust-region radius is still proportional to the optimality measure. Figures 3 and 4 depict these new results. The new `destress` method yields a better profile when compared to the standard trust-region approach on large ratios, while the efficiency gain can still be observed for small ratios. In view of such results, it appears that the use of a decoupling technique must be carefully thought, in order to both retain the characteristics on the traditional approach and improve upon them using decoupled steps when beneficial. For the trust-region framework, it seems that the parameters influencing the trust-region radii must be chosen according to the importance and tolerance related to each criterion.



(a) $\varepsilon_c = 10^{-3}$.



(b) $\varepsilon_c = 10^{-4}$.



(c) $\varepsilon_c = 10^{-6}$.

Figure 4: Performance of standard and decoupled (with alternative rule) trust-region methods given a budget of 10000 iterations.

5 Discussion

Let us focus this discussion on the derivative-based case. The result of Theorem 3.1 improves over the bound known [6] for a standard trust-region approach, which we recall is

$$\mathcal{O}\left(\max\left\{\varepsilon_{\mathtt{C}}^{-2}\varepsilon_{\mathtt{E}}^{-1}, \varepsilon_{\mathtt{E}}^{-3}\right\}\right).$$

For instance, such a bound is worse than (3.11) whenever $\varepsilon_{\mathtt{E}}^{-2} < \varepsilon_{\mathtt{C}}^{-2} < \varepsilon_{\mathtt{E}}^{-3}$, e.g., $\varepsilon_{\mathtt{C}} = 10^{-4}$ and $\varepsilon_{\mathtt{E}} = 10^{-3}$.

In terms of calls to the objective function and its derivatives, we observe that the iteration cost of Algorithm 2.1 can be comparable to that of a classical trust-region method enforcing second-order convergence. Indeed, such a scheme would also require the computation of two steps satisfying the same properties. If we compute our two models based on one gradient and one Hessian evaluation per iteration and thus set $m_k^{\mathtt{C}} = m_k^{\mathtt{E}}$, the method performs only one more function evaluation per iteration compared to classical second-order convergent frameworks (see [8] and [6]). Therefore, the evaluation complexity also benefits from our approach in terms of the dependence on the tolerances $\varepsilon_{\mathtt{C}}$ and $\varepsilon_{\mathtt{E}}$.

Our method can be viewed as a second-order decoupled variant of the trust-region method by Fan and Yuan [13]. We point out that the complexity of this algorithm was only analyzed in the general framework of nonlinear stepsize control, where worst-case complexity bounds of $\mathcal{O}\left(\varepsilon_{\mathtt{C}}^{-3}\right)$ for first-order optimality and $\mathcal{O}\left(\varepsilon_{\mathtt{E}}^{-3}\right)$ for a mixed criterion of first and second-order optimality were derived [17, 18]. Although we believe, as the authors of [17], that their first-order result can be improved to $\mathcal{O}\left(\varepsilon_{\mathtt{C}}^{-2}\right)$, we claim that the use of a decoupling technique is most likely necessary to achieve a bound as the one established Theorem 3.1.

We observe that the generic nonlinear stepsize control framework could be equipped with a decoupling phase, and that this would potentially lead to improved complexity results. In fact, many algorithms appear to be prone to "decoupling", and extensions of this concept to such schemes is an interesting perspective of the present work.

References

[1] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma. Finding approximate local minima for nonconvex optimization in linear time. arXiv:1611.01146v3, 2017.

[2] C. P. Avelino, J. M. Moguerza, A. Olivares, and F. J. Prieto. Combining and scaling descent and negative curvature directions. *Math. Program.*, 128:285–319, 2011.

[3] E. G. Birgin and J. M. Martínez. Quadratic regularization with cubic descent for unconstrained optimization. Technical Report MCDO271016, State University of Campinas, Campinas, Brazil, 2016.

[4] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for non-convex optimization. arXiv:1611.00756v2, 2017.

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

[6] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Complexity bounds for second-order optimality in unconstrained optimization. *J. Complexity*, 28:93–108, 2012.

[7] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Second-order optimality and beyond: characterization and evaluation complexity in convexly-constrained nonlinear optimization. Technical Report naXys-06-2016, Dept of Mathematics, UNamur, Namur (B), 2016.

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

[9] A. R. Conn, K. Scheinberg, and L. N. Vicente. Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM J. Optim.*, 20:387–415, 2009.

[10] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[11] F. E. Curtis, D. P. Robinson, and M. Samadi. A trust region algorithm with a worst-case iteration complexity of $O\left(\epsilon^{-3/2}\right)$ for nonconvex optimization. *Math. Program.*, 162:1–32, 2017.

[12] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[13] J. Fan and Y. Yuan. A new trust region algorithm with trust region radius converging to zero. In *Proceedings of the 5th International Conference on Optimization: Techniques and Applications*, Hong Kong, 2001.

[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] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Exploiting negative curvature directions in linesearch methods for unconstrained optimization. *Optim. Methods Softw.*, 14:75–98, 2000.

[16] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a Constrained and Unconstrained Testing Environment with safe threads. *Comput. Optim. Appl.*, 60:545–557, 2015.

[17] 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.

[18] G. N. Grapiglia, J. Yuan, and Y.-X. Yuan. Nonlinear stepsize control algorithms: Complexity bounds for first- and second-order optimality. *J. Optim. Theory Appl.*, 171:980–997, 2016.

[19] S. Gratton, C. W. Royer, and L. N. Vicente. A second-order globally convergent direct-search method and its worst-case complexity. *Optimization*, 65:1105–1128, 2016.

[20] D. Júdice. *Trust-Region Methods without using Derivatives: Worst Case Complexity and the Non-smooth Case*. PhD thesis, Dept. Mathematics, Univ. Coimbra, 2015.

[21] J. M. Martínez and M. Raydan. Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *J. Global Optim.*, 2017. doi: 10.1007/s10898-016-0475-8.

[22] J.J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Math. Program.*, 16:1–20, 1979.

[23] A. Olivares, J. M. Moguerza, and F. J. Prieto. Nonconvex optimization using negative curvature within a modified linesearch. *European J. Oper. Res.*, 189:706–722, 2008.

[24] G. A. Shultz, R. B. Schnabel, and R. H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM J. Numer. Anal.*, 22:47–67, 1985.

[25] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19:409–426, 1983.

[26] Y.-X. Yuan. Recent avances in trust region algorithms. *Math. Program.*, 151:249–281, 2015.