Projected proximal gradient trust-region algorithm for nonsmooth optimization

Minh N. Dao[*], Hung M. Phan[†], and Lindon Roberts[‡]

December 31, 2024

Abstract

We consider trust-region methods for solving optimization problems where the objective is the sum of a smooth, nonconvex function and a nonsmooth, convex regularizer. We extend the global convergence theory of such methods to include worst-case complexity bounds in the case of unbounded model Hessian growth, and introduce a new, simple nonsmooth trust-region subproblem solver based on combining several iterations of proximal gradient descent with a single projection into the trust region, which meets the sufficient descent requirements for algorithm convergence and has promising numerical results.

AMS Subject Classifications: 47H05; 49M37; 65K05; 65K10; 90C30.

Keywords: trust-region methods; nonsmooth optimization; proximal gradient descent; weak convexity.

1. Introduction

In this work, we consider trust-region methods for solving nonconvex, nonsmooth optimization problems of the form

$$\min_{x\in\mathbb{R}^d} F(x) := f(x) + h(x), \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is smooth (at least C^1) but nonconvex, and $h : \mathbb{R}^d \to (-\infty, +\infty]$ is a proper, lower semicontinuous and (possibly nonsmooth) convex function. The general form (1) encompasses convex-constrained optimization (where h is the indicator function for the constraint set) [HR22], and problems from areas such as optimal control of PDEs [HSW12] and imaging [ER21]. We assume that access to f is available through at least zeroth- and first-order oracles (i.e. at least f and ∇f are available to an algorithm) and access to h is available through direct evaluations and its proximity operator.

Most commonly, particularly in data science [WR22], first-order methods such as the proximal gradient method or FISTA [Bec17] are used to solve (1). In contrast, trust-region methods are typically second-order, using either $\nabla^2 f$ (if available) or a suitable quasi-Newton approximation

[*]School of Science, RMIT University, Melbourne, VIC 3000, Australia. E-mail: minh.dao@rmit.edu.au.

[†]Department of Mathematics and Statistics, Kennedy College of Sciences, University of Massachusetts Lowell, Lowell, MA 01854, USA. E-mail: hung_phan@uml.edu.

[‡]School of Mathematics and Statistics, University of Sydney, Camperdown NSW 2006, Australia. Email: lindon.roberts@sydney.edu.au.

such as symmetric rank-1 (SR1), and are well designed to exploit potential negative curvature in f [NW06].

In this work, we build on a recent trust-region method for solving (1) [BK23b]. The convergence analysis in [BK23b] allows for the model Hessians to grow unboundedly across iterations k, at a rate $O(k^t)$ as $k \to \infty$ for $t \in [0, 1]$. The uniformly bounded case $t = 0$ is the most widely used assumption for theoretical results (e.g. [CGT22]), but the case $t = 1$ can arise from using SR1 Hessian approximations [CGT00, Chapter 8.4.1.2], with convergence results in the smooth case (i.e. $h \equiv 0$ in (1)) derived in [Pow84]; the algorithm does not converge if $t > 1$ [Toi88]. By extending very recent results for smooth problems [DHO24], we broaden the convergence analysis from [BK23b] and show worst-case complexity bounds that match those of the smooth case. We also introduce a novel trust-region subproblem solver suitable for (1) based on a projected proximal gradient (PPG) iteration, and show that it meets the sufficient decrease requirements for the main trust-region algorithm. Our numerical results show that our PPG method can outperform existing subproblem solvers [BK23a].

1.1. Existing Works

In the special case where h is the indicator function for a convex constraint set, trust-region methods have been available for many years [CGT88, CGST93]. These methods achieve a worst-case complexity of $O(\epsilon^{-2})$ iterations to find an ϵ-approximate first-order critical point, although using cubic regularization methods can improve this to $O(\epsilon^{-3/2})$ [CGT12, CGT22], both in the case of uniformly bounded Hessians. Similarly, we note there are trust-region methods for general nonsmooth F, not exploiting the $F = f + h$ structure, such as [QS94, DLT95, CNY13, GJV16], and [CMW23] considers $F = f + h$ but builds a smooth approximation to F directly at each iteration. Alternative composite formulations have also been studied, such as $F = f + h \circ c$ for c smooth and nonconvex (and vector-valued) [GYY16] and $F = g \circ c + h$ for nonsmooth convex g, h and smooth nonconvex c [BE19, BE20], or (1) restricted to a smooth manifold [ZYZ24].

For the specific case of (1), proximal Newton methods extend the proximal gradient method to include second-order information about f by modifying the form of the proximity operator (potentially making it significantly harder to evaluate). This has allowed the development of second-order linesearch methods for (1) in the case of f both convex [LSS14] and nonconvex [LW19, KL21].

Trust-region methods for (1) but where h is nonconvex and nonsmooth are considered in [ABO22b, ABO22a, LO23]. These methods achieve worst-case complexity bounds of $O(\epsilon^{-2})$ iterations for uniformly bounded Hessians, and develop several subproblem solvers tailored to specific choices of regularizer h. By contrast, [KSD10] considers the case where h and f are both convex, yielding $O(\epsilon^{-1})$ worst-case complexity in this more restrictive setting. We also note [LLR24] considers the case where f is nonconvex but the local quadratic approximations used in the trust-region subproblem are convex (such as in some formulations of nonlinear least-squares problems), and the subproblem is solved with accelerated first-order methods.

The most relevant works for our results are [BK23b, BK23a, BK24]. These consider the same problem (1), albeit in an infinite-dimensional setting. In [BK23b], the general algorithmic framework is given, including sufficient decrease conditions for the trust-region subproblem. They show global convergence to first-order critical points under the assumption of potentially unbounded Hessians, specifically $\sum_{k=0}^{\infty}(1 + \max_{j\in\{0,\ldots,k\}} \|H_j\|)^{-1} = \infty$, where H_k is the model Hessian at iteration k, essentially assuming a growth rate of $\|H_k\| = O(k)$. Global convergence of smooth trust-region methods under this Hessian growth condition were shown in [Pow84] in the unconstrained case, and [Toi88] for the convex-constrained case. The theory from [Toi88] is extended to the case of general h in [BK23b], to give global convergence, and $O(\epsilon^{-2})$ complexity in the case of uniformly bounded model Hessians; this is extended to local convergence rates in [BK24] (with different local rates depending on the quality of Hessian approximation in the model for f).

A variety of trust-region subproblem solvers for (1) are also studied in [BK23b, BK23a]. These include nonsmooth generalizations of the standard Cauchy point and dogleg method for the smooth ($h \equiv 0$) case and a nonlinear conjugate gradient method. However, from the numerical results in [BK23a], the most successful method developed in these works is the so-called spectral proximal gradient (SPG) method. This method essentially performs proximal gradient iterations with some degree of linesearch to enforce both sufficient decrease and feasibility with respect to the trust region constraint.

Lastly, we note that both [KSD10] and [OM24] consider trust-region subproblem solvers for (1) based on combining an iterative solver with a final projection step, which is similar to the approach in our new subproblem solver. However, [KSD10] considers only the case where f is convex, and [OM24] uses a semismooth Newton formulation.

1.2. Contributions

We provide two main contributions in this work.

Firstly, we build on the theoretical analysis of the general nonsmooth trust-region method from [BK23b] to include worst-case complexity in the case of unbounded model Hessians. Broadly speaking, if the model Hessians grow at a rate $\|H_k\| = O(k^t)$ for $t \in [0, 1]$, then we prove convergence to an ϵ-approximate first-order critical point after at most $O(\epsilon^{-2/(1-t)})$ iterations if $t \in [0, 1)$, and $\tilde{O}(e^{-c\epsilon^{-2}})$ iterations (for some $c > 0$) if $t = 1$. Our analysis is based on recent complexity results for the smooth case $h \equiv 0$ [DHO24]; our worst-case complexity bounds in the nonsmooth case are identical to the smooth case.

Secondly, we introduce a new trust-region subproblem solver suitable for use in the main algorithm, which we call a *projected proximal gradient (PPG) method*. Our approach is simpler than the SPG method from [BK23b, BK23a] in that we perform several iterations of proximal gradient descent on our model, and only once, at the end of our iterations, compute a projection to satisfy the trust region constraint. We show that this method satisfies the sufficient decrease conditions required for global convergence (and complexity bounds) of the main trust-region algorithm. We numerically compare our new PPG method against SPG on a large collection of ℓ_1-regularized CUTEst [GOT15] problems, and show that PPG can outperform SPG (in the sense of the whole trust-region algorithm converging more quickly) when high accuracy solutions are desired. We also observe that PPG is better able to make use of larger subproblem iteration budget if available.

1.3. Organization and Notation

In Section 2, we provide the relevant technical background for nonsmooth functions. The main trust-region algorithm and its worst-case complexity in the unbounded Hessian case are given in Section 3. In Section 4, we introduce our new subproblem solver, and we present our numerical results in Section 5.

Throughout, we use $\|\cdot\|$ to represent the Euclidean norm in $\mathbb{R}^d$ and the matrix 2-norm.

2. Preliminaries

Given $h\colon \mathbb{R}^d \to (-\infty, +\infty]$, its *domain* is $\operatorname{dom} h := \{x \in \mathbb{R}^d : h(x) < +\infty\}$ and its *epigraph* is $\operatorname{epi} h := \{(x, \rho) \in \mathbb{R}^d \times \mathbb{R} : \rho \geq h(x)\}$. We recall that h is *proper* if $\operatorname{dom} h \neq \varnothing$, *lower semicontinuous* if $\operatorname{epi} h$ is a closed set, and *convex* if $\operatorname{epi} h$ is a convex set.

Let $h\colon \mathbb{R}^d \to (-\infty, +\infty]$ be proper. The *regular subdifferential* of h at $x \in \mathbb{R}^d$ is defined by

$$\partial h(x) := \left\{ u \in \mathbb{R}^d : \liminf_{y \to \mathbb{R}^d} \frac{h(y) - h(x) - u^\top (y - x)}{\|y - x\|} \geq 0 \right\}, \tag{2}$$

if $x \in \operatorname{dom} h$, and $\partial h(x) := \varnothing$ otherwise. When h is convex, the regular subdifferential coincides with the *convex subdifferential*, see, e.g., [Mor06, Theorem 1.93],

$$\partial h(x) = \{u \in \mathbb{R}^d : \forall y \in \mathbb{R}^d,\ h(x) + u^\top (y - x) \leq h(y)\}. \tag{3}$$

Proposition 2.1 ([Mor06, Proposition 1.107(i)]). *Let $h\colon \mathbb{R}^d \to (-\infty, +\infty]$ be proper and $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ be differentiable at $x \in \operatorname{dom} h$. Then*

$$\partial (f + h)(x) = \nabla f(x) + \partial h(x). \tag{4}$$

We recall that the *proximity operator* of h with parameter $\gamma > 0$ at $x \in \mathbb{R}^d$ is

$$\operatorname{Prox}_{\gamma h}(x) := \operatorname*{argmin}_{z \in \mathbb{R}^d} \left(h(z) + \frac{1}{2\gamma} \|z - x\|^2 \right). \tag{5}$$

Proposition 2.2. *Let $h\colon \mathbb{R}^d \to (-\infty, +\infty]$ be a lower semicontinuous convex function. Then*

(i) *$\operatorname{Prox}_{\gamma h}$ is single-valued, and $u = \operatorname{Prox}_{\gamma h}(x)$ if and only if $x - u \in \gamma \partial h(u)$.*
(ii) *$\operatorname{Prox}_{\gamma h}$ is nonexpansive, i.e., for all $x, y \in \mathbb{R}^d$, $\|\operatorname{Prox}_{\gamma h}(x) - \operatorname{Prox}_{\gamma h}(y)\| \leq \|x - y\|$.*

Proof. See, e.g. [Bec17, Theorems 6.39 & 6.42]. ∎

For problem (1), we will assume f is differentiable and we have access to the proximity operator for h (see Assumption 3.1 below), and so consider the first-order stationarity measure given by

$$\pi(x, \gamma) := \frac{1}{\gamma} \|\operatorname{Prox}_{\gamma h}(x - \gamma \nabla f(x)) - x\|, \tag{6}$$

for any $x \in \mathbb{R}^d$ and $\gamma > 0$, which is the same measure considered in [BK23a]. When we consider our main algorithm and generate iterates x_k, we will use the shorthand $\pi_k(\gamma) := \pi(x_k, \gamma)$.

The function $\pi(x, \gamma)$ has several useful properties for a stationarity measure, such as $\pi(x, \gamma) \geq 0$ with $\pi(x, \gamma) = 0$ if and only if $0 \in \nabla f(x) + \partial h(x)$, and being continuous in both x and γ [BK23a, Proposition 1].

Lemma 2.3 ([Bec17, Theorem 10.9]). *For any $x \in \mathbb{R}^d$ and $0 < \gamma_1 \leq \gamma_2$,*

$$\pi(x, \gamma_2) \leq \pi(x, \gamma_1) \leq \frac{\gamma_2}{\gamma_1} \pi(x, \gamma_2). \tag{7}$$

For $\lambda \in \mathbb{R}$, a function $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ is said to be λ-*convex* if for all $x, y \in \operatorname{dom} f$ and all $\alpha \in (0,1)$,

$$f((1-\alpha)x + \alpha y) + \frac{\lambda}{2}\alpha(1-\alpha)\|x-y\|^2 \leq (1-\alpha)f(x) + \alpha f(y). \tag{8}$$

It is well known that f is λ-convex if and only if $f - \frac{\lambda}{2}\|\cdot\|^2$ is convex, see, e.g., [DP19, Section 5]. Then we say that f is *strongly convex* if $\lambda > 0$, and *weakly convex* if $\lambda < 0$. More properties of λ-convex functions can be found in, e.g., [DP19, Section 5].

Example 2.4. Recall that for a symmetric matrix $H \in \mathbb{R}^{d\times d}$, we have $H \succeq \lambda_{\min} I$ where $\lambda_{\min}$ is the minimum eigenvalue of H.

(i) The quadratic function $p \mapsto \frac{1}{2}p^\top H p$ is $\lambda_{\min}$-convex. In addition, we note that $\|H\| = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } H\}$. Thus,

$$\|H\| \geq \lambda_{\min}. \tag{9}$$

(ii) If f_1 is λ_1-convex and f_2 is λ_2-convex, then $f_1 + f_2$ is $(\lambda_1 + \lambda_2)$-convex, see, e.g., [**?**, Lemma 5.3]. Thus, for any $x \in \mathbb{R}^d$, $g \in \mathbb{R}^d$, and any convex function h, the function

$$m(p) := c + g^\top p + \frac{1}{2}p^\top H p + h(x+p), \tag{10}$$

is also $\lambda_{\min}$-convex.

Proposition 2.5. *Let $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ be proper, lower semicontinuous, and λ-convex. Then, for all $x, y \in \mathbb{R}^d$ and all $u \in \partial f(x)$,*

$$f(y) - f(x) \geq u^\top (y-x) + \frac{\lambda}{2}\|y-x\|^2. \tag{11}$$

Proof. This follows from [DPT24, Proposition 2.1(ii)]. ■

3. Trust-Region Algorithm and Worst-Case Complexity

We begin by outlining our main trust-region algorithm for solving (1), in the case of Assumption 3.1. This algorithm is essentially the same as [BK23a, Algorithm 1], but we will extend the analysis from global convergence to worst-case complexity, following the approach from [DHO24]. Our algorithm is designed to solve problems of the form (1) satisfying the following conditions.

Assumption 3.1.

(i) The smooth component f is differentiable with ∇f being $L_{\nabla f}$-Lipschitz continuous.
(ii) The nonsmooth component h is proper, lower semicontinuous and convex.
(iii) The whole objective F is bounded below by F_{low}.

Although not a formal assumption, our algorithm is designed using the proximity operator of h, and so we assume $\operatorname{Prox}_{\gamma h}$ is practical to compute. See [Bec17, Chapter 6.9] for examples of such functions.

Algorithm 1 Nonsmooth trust-region method for (1).

```
Input: Starting point $x_0 \in \operatorname{dom} F$ and trust-region radius $\Delta_0 > 0$. Algorithm parameters: scaling
    factors $0 < \gamma_{\text{dec}} < 1 < \gamma_{\text{inc}}$ and acceptance threshold $\eta \in (0,1)$.
 1: for $k = 0, 1, 2, \ldots$ do
 2:     Build a local quadratic model $m_k$ (12) satisfying Assumption 3.2.
 3:     Solve the trust-region subproblem (13) to get a step $p_k$ satisfying Assumption 3.3.
 4:     Evaluate $F(x_k + p_k)$ and calculate the ratio
            $$\rho_k := \frac{F(x_k) - F(x_k + p_k)}{m_k(0) - m_k(p_k)}.  \qquad (14)$$
 5:     if $\rho_k \geq \eta$ then
 6:         (Successful iteration) Set $x_{k+1} = x_k + p_k$ and $\Delta_{k+1} = \gamma_{\text{inc}} \Delta_k$.
 7:     else
 8:         (Unsuccessful iteration) Set $x_{k+1} = x_k$ and $\Delta_{k+1} = \gamma_{\text{dec}} \Delta_k$.
 9:     end if
10: end for
```

3.1. Main Algorithm

As is standard in trust-region methods [CGT00], at each iteration k (with current iterate x_k), we construct a local approximation to the full objective F (1). In our case, we approximate the smooth part f with a quadratic approximation and assume exact access to h, yielding an approximation

$$F(x_k + p) \approx m_k(p) := f(x_k) + \nabla f(x_k)^\top p + \frac{1}{2} p^\top H_k p + h(x_k + p), \tag{12}$$

where $H_k \in \mathbb{R}^{d \times d}$ is some symmetric matrix approximating the curvature of f, for example via symmetric rank-1 quasi-Newton updating [NW06, Chapter 6.2]. We then compute a tentative step p_k by approximately minimizing the model subject to a constraint on the size of the step, the trust-region subproblem

$$p_k \approx \operatorname*{argmin}_{p \in \mathbb{R}^d} m_k(p) \quad \text{s.t.} \quad \|p\| \leq \Delta_k, \tag{13}$$

for some value $\Delta_k > 0$ which is dynamically updated by the algorithm. Lastly, we decide whether to accept the tentative new iterate $x_k + p_k$ by computing the ratio of the actual decrease in the objective from the step p_k compared to the predicted decrease implied by the model (14). This full procedure is given in Algorithm 1.

As in [BK23b, BK23a], we assume that our model Hessian H_k may grow unboundedly. Specifically, we follow [DHO24, Model Assumption 4.1] and (broadly speaking) assume that $\|H_k\| = O(k^t)$ for some $t \in [0, 1]$. The case $t = 0$ corresponds to the more common assumption of uniformly bounded Hessians, but $t = 1$ can occur when H_k is generated by symmetric rank-1 quasi-Newton updating [CGT00, Chapter 8.4.1.2]. If $t > 1$, the algorithm can be shown to not be globally convergent [Toi88].

Assumption 3.2. There exist $\mu > 0$ and $t \in [0, 1]$ such that, for all $k \in \mathbb{N}$,

$$\max_{j \in \{0, \ldots, k\}} \|H_j\| \leq \mu(1 + k^t). \tag{15}$$

Lastly, we again follow [BK23a] and require the following sufficient decrease condition from the (approximate) trust-region step (13).

Assumption 3.3. There exists $\kappa_p > 0$ and $\gamma_{\max} > 0$ (both independent of k) such that for all iterations $k \in \mathbb{N}$, the step p_k satisfies

$$m_k(0) - m_k(p_k) \geq \kappa_p \pi_k(\gamma_{\max}) \min\left\{\Delta_k, \frac{\pi_k(\gamma_{\max})}{\|H_k\| + 1}\right\}, \tag{16}$$

where $\pi_k(\gamma) := \pi(x_k, \gamma)$ is the first-order stationarity measure at x_k.

A key feature of this work is the new *projected proximal gradient subproblem solver* introduced in Section 4. In Corollary 4.4, we will show that this solver satisfies Assumption 3.3. Alternatively, [BK23a] provides several suitable subproblem solvers.

The global convergence of Algorithm 1 has already been established.

Theorem 3.4. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. Then*

$$\liminf_{k\to\infty} \pi_k(\gamma_{\max}) = 0. \tag{17}$$

Proof. This is [BK23a, Theorem 1], which is a small extension of [BK23b, Theorem 3]. ■

3.2. Worst-Case Complexity

We now extend the analysis of Algorithm 1 beyond the global convergence in Theorem 3.4 and consider its worst-case complexity. That is, we wish to bound the number of iterations it takes to drive the stationarity measure $\pi_k(\gamma_{\max})$ below some desired level $\epsilon > 0$. The analysis in this section follows [DHO24], which considers only the smooth case $h \equiv 0$.

First, we bound the error in our model (12). Denote

$$c_1 := \frac{1}{2}\max(L_{\nabla f}, 1). \tag{18}$$

Lemma 3.5. *Suppose Assumption 3.1 holds. Then the trust-region step p_k satisfies*

$$|F(x_k + p_k) - m_k(p_k)| \leq c_1(1 + \|H_k\|)\Delta_k^2, \tag{19}$$

where c_1 is given by (18).

Proof. Since ∇f is $L_{\nabla f}$-Lipschitz continuous, we have the standard bound (e.g. [NW06, Appendix A])

$$|f(x_k + p_k) - f(x_k) - \nabla f(x_k)^\top p_k| \leq \frac{1}{2} L_{\nabla f}\|p_k\|^2. \tag{20}$$

Then by the definitions of F and m_k we have

$$\begin{aligned} |F(x_k + p_k) - m_k(p_k)| &= |f(x_k + p_k) + h(x_k + p_k) \\ &\qquad - f(x_k) - \nabla f(x_k)^\top p_k - \frac{1}{2}p_k^\top H_k p_k - h(x_k + p_k)|, && (21) \\ &\leq \frac{1}{2}L_{\nabla f}\|p_k\|^2 + \frac{1}{2}\|H_k\| \cdot \|p_k\|^2, && (22) \\ &\leq c_1\|p_k\|^2 + c_1\|H_k\| \cdot \|p_k\|^2. && (23) \end{aligned}$$

The result then follows from $\|p_k\| \leq \Delta_k$. ■

An important quantity to track in our analysis is

$$a_k := \Delta_k \cdot \frac{1 + \max_{j \le k} \|H_j\|}{\min_{j \le k} \pi_j(\gamma_{\max})}. \tag{24}$$

We have the following.

Lemma 3.6. *Suppose Assumptions 3.1 and 3.3 hold. Then, for all $k \in \mathbb{N}$,*

$$a_k \ge a_{\min} := \min\left\{a_0, \gamma_{\text{dec}}, \frac{\gamma_{\text{dec}}\kappa_p(1-\eta)}{c_1}\right\}, \tag{25}$$

where a_k and c_1 are from (24) and (18), respectively.

Proof. We first note that a_k/Δ_k is non-decreasing by definition, and so

$$\frac{a_{k+1}}{\Delta_{k+1}} \ge \frac{a_k}{\Delta_k}. \tag{26}$$

We will show the result by induction. The case $k = 0$ is trivial by the definition of $a_{\min}$, so now suppose $a_k \ge a_{\min}$ for some k.

First, if $a_k \ge \min\{1, \kappa_p(1-\eta)/c_1\}$, then by the updating mechanism for Δ_k (see Algorithm 1) and (26) we have

$$a_{k+1} = \frac{a_{k+1}}{\Delta_{k+1}} \cdot \Delta_{k+1} \ge \frac{a_k}{\Delta_k} \cdot \gamma_{\text{dec}}\Delta_k = \gamma_{\text{dec}} a_k \ge \gamma_{\text{dec}} \min\left\{1, \frac{\kappa_p(1-\eta)}{c_1}\right\} \ge a_{\min}, \tag{27}$$

and we are done.

Now suppose that $a_k < \min\{1, \kappa_p(1-\eta)/c_1\}$. Then since $m_k(0) = F(x_k)$ by the definition of m_k, we have

$$|\rho_k - 1| = \frac{|F(x_k + p_k) - m_k(p_k)|}{m_k(0) - m_k(p_k)} \le \frac{c_1(1 + \|H_k\|)\Delta_k^2}{\kappa_p \pi_k(\gamma_{\max}) \min\left\{\Delta_k, \frac{\pi_k(\gamma_{\max})}{1+\|H_k\|}\right\}}, \tag{28}$$

using Lemma 3.5 and Assumption 3.3. Hence

$$\begin{aligned} |\rho_k - 1| &\le \frac{c_1\left(1 + \max_{j \le k}\|H_j\|\right)\Delta_k^2}{\kappa_p\left[\min_{j \le k}\pi_j(\gamma_{\max})\right]\min\left\{\Delta_k, \frac{\min_{j\le k}\pi_j(\gamma_{\max})}{1+\max_{j\le k}\|H_j\|}\right\}}, &(29)\\ &= \frac{c_1 a_k \Delta_k}{\kappa_p \min\left\{\Delta_k, \frac{\Delta_k}{a_k}\right\}}, &(30)\\ &= \frac{c_1}{\kappa} a_k < 1 - \eta, &(31)\end{aligned}$$

using the assumption $a_k < 1$ to get $\min(\Delta_k, \Delta_k/a_k) = \Delta_k$ for the last equality. Thus we have $\rho_k \ge \eta$, so iteration k is successful. By the trust-region updating mechanism, $\Delta_{k+1} \ge \Delta_k$, and so using (26) we get

$$a_{k+1} = \frac{a_{k+1}}{\Delta_{k+1}} \cdot \Delta_{k+1} \ge \frac{a_k}{\Delta_k} \cdot \Delta_k = a_k \ge a_{\min}, \tag{32}$$

and we are done. ∎

Lemma 3.7. *Suppose Assumptions 3.1 and 3.3 hold. Then for each k,*

$$m_k(0) - m_k(p_k) \geq \kappa_p a_{\min} \cdot \frac{\min_{j\leq k} \pi_j(\gamma_{\max})^2}{1 + \max_{j\leq k} \|H_j\|}, \tag{33}$$

where $a_{\min}$ is defined in (25).

Proof. From Assumption 3.3 and the definition (24) of a_k, we have

$$\begin{aligned} m_k(0) - m_k(p_k) &\geq \kappa_p \pi_k(\gamma_{\max}) \min\left\{\Delta_k, \frac{\pi_k(\gamma_{\max})}{1+\|H_k\|}\right\}, & (34)\\ &\geq \kappa_p \left[\min_{j\leq k} \pi_j(\gamma_{\max})\right] \min\left\{\frac{a_k \cdot \min_{j\leq k}\pi_j(\gamma_{\max})}{1+\max_{j\leq k}\|H_j\|}, \frac{\min_{j\leq k}\pi_j(\gamma_{\max})}{1+\max_{j\leq k}\|H_j\|}\right\}, & (35)\\ &= \kappa_p \cdot \min\{a_k, 1\} \cdot \frac{\min_{j\leq k}\pi_j(\gamma_{\max})^2}{1+\max_{j\leq k}\|H_j\|}. & (36)\end{aligned}$$

The result then follows from $a_k \geq a_{\min}$ (Lemma 3.6) and $1 > \gamma_{\text{dec}} \geq a_{\min}$. ∎

We also need the following technical lemmas. The first is relevant to Assumption 3.2, and the second will be used to bound the time before the first successful iteration.

Lemma 3.8. *Suppose $\mu > 0$ and $t > 0$. Then for all $k_1 < k_2$ we have*

$$\sum_{k=k_1}^{k_2} \frac{1}{1+\mu(1+(k+1)^t)} \geq \frac{(k_1+1)^t}{1+\mu(1+(k_1+1)^t)} \int_{k_1+1}^{k_2+2} \frac{1}{s^t} ds. \tag{37}$$

Proof. This is [DHO24, Lemma 5]. ∎

Lemma 3.9. *Suppose $a_1, a_2 > 0$. Then there exists k^* such that $k > k^*$ implies $k > a_1 + a_2\sqrt{k}$, and moreover we have $k^* = \Theta(a_1)$ as $a_1 \to \infty$.*

Proof. The inequality $k \geq a_1 + a_2\sqrt{k}$ holds provided that $\sqrt{k}$ is at least the larger root of $t^2 - a_2 t - a_1$, i.e,

$$k > \left(\frac{a_2 + \sqrt{a_2^2 + 4a_1}}{2}\right)^2 = \frac{2a_1 + a_2^2 + a_2\sqrt{a_2^2+4a_1}}{2}, \tag{38}$$

which gives $k^* = \Theta(a_1)$. ∎

Our first main result bounds the total time until the first successful iteration.

Lemma 3.10. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. Then if $\pi_0(\gamma_{\max}) \geq \epsilon > 0$, then there is at least one successful iteration, and if k_0 is the first successful iteration we have $k_0 = O(\log(1/\epsilon))$ as $\epsilon \to 0$.*

Proof. We have that the iterations $k = 0, \ldots, k_0 - 1$ are unsuccessful, and so $x_0 = x_1 = \cdots = x_{k_0}$. Thus, $\pi_0(\gamma_{\max}) = \cdots = \pi_{k_0}(\gamma_{\max}) \geq \epsilon$ and $\Delta_{k_0} = \gamma_{\text{dec}}^{k_0}\Delta_0$. Applying Lemma 3.6 and Assumption 3.2, we have (for all k)

$$\gamma_{\text{dec}}^{k_0}\Delta_0 = \Delta_{k_0} \geq \frac{\min_{j\leq k_0}\pi_j(\gamma_{\max})}{1+\max_{j\leq k_0}\|H_j\|} \cdot a_{\min} \geq \frac{\epsilon}{1+\mu(1+k_0^t)} \cdot a_{\min}. \tag{39}$$

Equivalently,

$$\log(1+\mu(1+k_0^t)) \geq \log\left(\frac{\epsilon a_{\min}}{\Delta_0}\right) + k_0 \log(\gamma_{\text{dec}}^{-1}). \tag{40}$$

Using the identity $\log(x) \leq \frac{2}{e}\sqrt{x}$ for all $x > 0$ [Mit70, Section 3.6.1], together with $t \leq 1$ we get

$$\frac{2}{e}\sqrt{1+\mu+\mu k_0} \geq \log\left(\frac{\epsilon a_{\min}}{\Delta_0}\right) + k_0 \log(\gamma_{\text{dec}}^{-1}). \tag{41}$$

Now using the identity $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ for $x, y \geq 0$, we have

$$\frac{2\sqrt{1+\mu}}{e} + \frac{2\sqrt{\mu}}{e}\sqrt{k_0} \geq \log\left(\frac{\epsilon a_{\min}}{\Delta_0}\right) + k_0 \log(\gamma_{\text{dec}}^{-1}), \tag{42}$$

or

$$k_0 \leq \frac{1}{\log(\gamma_{\text{dec}}^{-1})}\left[\frac{2\sqrt{1+\mu}}{e} + \log\left(\frac{\Delta_0}{\epsilon a_{\min}}\right)\right] + \frac{2\sqrt{\mu}}{e\log(\gamma_{\text{dec}}^{-1})}\sqrt{k_0}. \tag{43}$$

From Lemma 3.9, we get that k_0 is finite, and $k_0 \leq O(\log(1/\epsilon))$ as $\epsilon \to 0$. ■

Now consider a desired accuracy level ϵ, and define k_ϵ to be the first iteration with $\pi_{k_\epsilon}(\gamma_{\max}) < \epsilon$ (or $k_\epsilon = \infty$ if this never occurs).

If $\pi_0(\gamma_{\max}) < \epsilon$, then we have $k_\epsilon = 0$ and no further effort is required. Hence we will assume our desired accuracy level satisfies $\epsilon \leq \pi_0(\gamma_{\max})$. In this case, from Lemma 3.10 we have a first successful iteration k_0 with $k_0 = O(\log(1/\epsilon))$. Since x_k is only changed on successful iterations, we have $\pi_k(\gamma_{\max}) = \pi_0(\gamma_{\max})$ for all $k \leq k_0$, and so we must have $k_\epsilon > k_0$.

Assume that

$$\tau \text{ is a positive integer such that } \gamma_{\text{inc}}\gamma_{\text{dec}}^{\tau-1} < 1, \tag{44}$$

which must exist due to $\gamma_{\text{dec}} < 1$.

We now define the following sets:

- $\mathcal{S} := \{j \in \{0,1,2,\dots\} : \rho_j \geq \eta\}$ is the set of all successful iterations
- $\mathcal{S}_k := \{j \in \{0,1,2,\dots,k\} : \rho_j \geq \eta\} = \{j \in \{k_0,\dots,k\} : \rho_j \geq \eta\}$ is the set of all successful iterations up to iteration k, with $\mathcal{S}(\epsilon) := \mathcal{S}_{k_\epsilon}$.
- $\mathcal{U}_k := \{j \in \{0,1,2,\dots,k\} : \rho_j < \eta\}$ is the set of all unsuccessful iterations up to iteration k, with $\mathcal{U}(\epsilon) := \mathcal{U}_{k_\epsilon}$.
- For $k \geq k_0$, define $\mathcal{V}_k := \{j \in \{k_0,\dots,k\} : |\mathcal{S}_j| \geq j/\tau\}$ is the set of iterations for which we have had a relatively high fraction of successful iterations in the history to that point.[1]
- For $k \geq k_0$, define $\mathcal{W}_k := \{j \in \{k_0,\dots,k\} : |\mathcal{S}_j| < j/\tau\}$ is the set of iterations for which we have had a relatively low fraction of successful iterations, $\mathcal{W}_k = \{k_0,\dots,k\} \setminus \mathcal{V}_k$.

[1]For example, if $\tau = 3$ and $j \in \mathcal{V}_k$, then at least 1/3 of the iterations $0,\dots,j$ were successful. We may choose $\tau = 3$ if for example we have the common parameter choices $\gamma_{\text{inc}} = 2$ and $\gamma_{\text{dec}} = 0.5$.

Lemma 3.11. *Let r_k be any strictly positive, non-decreasing sequence, and we have at least one successful iteration (i.e. k_0 exists). Then for any $k \geq k_0$, we have*

$$\tau \sum_{j\in\mathcal{S}_k} \frac{1}{r_j} \geq \sum_{j\in\mathcal{V}_k} \frac{1}{r_j} = \sum_{j=k_0}^{k} \frac{1}{r_j} - \sum_{j\in\mathcal{W}_k} \frac{1}{r_j}, \tag{45}$$

using the convention $\sum_{j\in\emptyset} \frac{1}{r_j} = 0$.

Proof. This is [DHO24, Lemma 7].

■

Lemma 3.12. *Suppose Assumptions 3.1, 3.2 and 3.3 hold, and we have at least one successful iteration (i.e. k_0 exists). Then for any $\epsilon > 0$,*

$$\sum_{k\in\mathcal{W}_{k_\epsilon}} \frac{1}{1+\mu(1+k^t)} \leq \frac{\Delta_0 \xi}{\epsilon a_{\min}}, \tag{46}$$

again with the convention $\sum_{k\in\emptyset} a_k = 0$, and where

$$\xi := \sum_{k\in\mathbb{N}} (\gamma_{\text{inc}} \gamma_{\text{dec}}^{\tau-1})^{k/\tau} < \infty. \tag{47}$$

Proof. First, note that indeed $\xi < \infty$ since it is a geometric series $\sum_{k\in\mathbb{N}} \delta^k$ for $\delta = (\gamma_{\text{inc}}\gamma_{\text{dec}}^{\tau-1})^{1/\tau}$ with $\delta < 1$ by the definition of τ in (44).

If $\mathcal{W}_{k_\epsilon} = \emptyset$ the result is trivial. Otherwise, fix $k \in \mathcal{W}_{k_\epsilon}$ (and so in particular $k \leq k_\epsilon$) and let $r_k = 1+\mu(1+k^t)$. Then Assumption 3.2, Lemma 3.6 and the update mechanism for Δ_k give

$$\frac{a_{\min}\epsilon}{r_k} \leq a_k \frac{\min_{j\leq k} \pi_j(\gamma_{\max})}{1+\max_{j\leq k} \|H_j\|} \leq \Delta_k = \gamma_{\text{inc}}^{|\mathcal{S}_k|} \gamma_{\text{dec}}^{k-|\mathcal{S}_k|} \Delta_0. \tag{48}$$

Since $k \in \mathcal{W}_{k_\epsilon}$, we have $k > \tau|\mathcal{S}_k|$, and so

$$\frac{a_{\min}\epsilon}{r_k} \leq \gamma_{\text{inc}}^{k/\tau} \gamma_{\text{dec}}^{k-k/\tau} \Delta_0 = (\gamma_{\text{inc}}\gamma_{\text{dec}}^{\tau-1})^{k/\tau} \Delta_0. \tag{49}$$

Summing over all $k \in \mathcal{W}_{k_\epsilon}$ then gives

$$a_{\min}\epsilon \sum_{k\in\mathcal{W}_{k_\epsilon}} \frac{1}{r_k} \leq \sum_{k\in\mathcal{W}_{k_\epsilon}} (\gamma_{\text{inc}}\gamma_{\text{dec}}^{\tau-1})^{k/\tau}\Delta_0 \leq \sum_{k\in\mathbb{N}} (\gamma_{\text{inc}}\gamma_{\text{dec}}^{\tau-1})^{k/\tau}\Delta_0, \tag{50}$$

and we are done. ■

We can now state our main worst-case complexity result.

Theorem 3.13. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. Then for any $0 < \epsilon \leq \pi_0(\gamma_{\max})$, if $0 \leq t < 1$ we have*

$$k_\epsilon \leq \left[(1-t)(c_2\epsilon^{-2} + c_3\epsilon^{-1}) \frac{1+\mu(1+k_0^t)}{1+k_0^t} + (k_0+1)^{1-t}\right]^{1/(1-t)} - 2, \tag{51}$$

and if $t = 1$ we have

$$k_\epsilon \leq \exp\left((c_2\epsilon^{-2} + c_3\epsilon^{-1})\frac{1+\mu(1+k_0^t)}{1+k_0^t}\right)(k_0+1) - 2, \tag{52}$$

where in both bounds we use the constants

$$c_2 := \frac{\tau[F(x_0) - F_{\text{low}}]}{\eta\kappa a_{\min}}, \qquad \textit{and} \qquad c_3 := \frac{\Delta_0\xi}{a_{\min}}. \tag{53}$$

If instead $\epsilon > \pi_0(\gamma_{\max})$, then $k_\epsilon = 0$.

Proof. If $\epsilon > \pi_0(\gamma_{\max})$, the result is trivial. Otherwise, from $\epsilon \leq \pi_0(\gamma_{\max})$ and Lemma 3.10, we know that there must be a first successful iteration (i.e. k_0 exists).

For successful iterations $k \in \mathcal{S}(\epsilon)$, we have, using Lemma 3.7,

$$F(x_k) - F(x_k + p_k) \geq \eta(m_k(0) - m_k(p_k)), \tag{54}$$

$$\geq \eta\kappa a_{\min} \cdot \frac{\min_{j\leq k}\pi_j(\gamma_{\max})^2}{1 + \max_{j\leq k}\|H_j\|}, \tag{55}$$

$$\geq \eta\kappa a_{\min} \cdot \frac{\epsilon^2}{1+\mu(1+k^t)}. \tag{56}$$

Summing over all $k \in \mathcal{S}(\epsilon)$ (and noting that $x_{k+1} = x_k$ for $k \notin \mathcal{S}(\epsilon)$), we have

$$F(x_0) - F_{\text{low}} \geq \sum_{k\in\mathcal{S}(\epsilon)} F(x_k) - F(x_k+p_k) \geq \eta\kappa a_{\min}\epsilon^2 \sum_{k\in\mathcal{S}(\epsilon)} \frac{1}{1+\mu(1+k^t)}. \tag{57}$$

Define $r_k = 1 + \mu(1+k^t)$, which is strictly positive and non-increasing. Hence Lemmas 3.11 and 3.12 give

$$F(x_0) - F_{\text{low}} \geq \frac{\eta\kappa a_{\min}\epsilon^2}{\tau}\left[\sum_{j=k_0}^{k_\epsilon}\frac{1}{r_j} - \sum_{j\in\mathcal{W}_{k_\epsilon}}\frac{1}{r_j}\right], \tag{58}$$

$$\geq \frac{\eta\kappa a_{\min}\epsilon^2}{\tau}\left[\sum_{j=k_0}^{k_\epsilon}\frac{1}{r_j} - \frac{\Delta_0\xi}{\epsilon a_{\min}}\right]. \tag{59}$$

Finally, Lemma 3.8 gives

$$F(x_0) - F_{\text{low}} \geq \frac{\eta\kappa a_{\min}\epsilon^2}{\tau}\left[\frac{(1+k_0^t)}{r_{k_0}}\int_{k_0+1}^{k_\epsilon+2}\frac{1}{s^t}ds - \frac{\Delta_0\xi}{\epsilon a_{\min}}\right], \tag{60}$$

or equivalently

$$\int_{k_0+1}^{k_\epsilon+2}\frac{1}{s^t}ds \leq \left(c_2\epsilon^{-2} + c_3\epsilon^{-1}\right)\frac{r_{k_0}}{1+k_0^t}. \tag{61}$$

We get the final result from evaluating the integral,

$$\int_{k_0+1}^{k_\epsilon+2}\frac{1}{s^t}ds = \begin{cases}\frac{(k_\epsilon+2)^{1-t}-(k_0+1)^{1-t}}{1-t}, & \text{if } 0 \leq t < 1,\\ \log(k_\epsilon+2) - \log(k_0+1), & \text{if } t = 1,\end{cases} \tag{62}$$

and rearranging for k_ϵ. ■

Algorithm 2 Projected proximal gradient method for nonsmooth trust-region subproblem (63).

```
    Parameters: stepsize $\gamma > 0$, maximum iterations $N \in \mathbb{N}$ and trust-region scaling $\mu_u \geq 1$.
1:  Set $i = 0$ and $u_0 = x$.
2:  while $i < N$ and $\|u_i - x\| \leq \mu_u \Delta$ do
3:      $u_{i+1} = \text{Prox}_{\gamma h}(u_i - \gamma g - \gamma H(u_i - x))$.
4:      $i = i + 1$.
5:  end while
6:  return $p^* = \frac{\Delta}{\max\{\Delta, \|u_i - x\|\}}(u_i - x)$.
```

In summary, the worst-case complexity bounds are the same as the smooth case [DHO24], as summarized below. In the case of uniformly bounded Hessians ($t = 0$), we recover the standard $O(\epsilon^{-2})$ worst-case iteration complexity for smooth trust-region methods (e.g. [CGT22, Theorem 2.3.7]).

Corollary 3.14. *Suppose the assumptions of Theorem 3.13 hold. If $0 \leq t < 1$, then $k_\epsilon = O(\epsilon^{-2/(1-t)})$, and if $t = 1$ then $k_\epsilon = \tilde{O}(e^{c\epsilon^{-2}})$ for some $c > 0$, where $\tilde{O}(\cdot)$ hides logarithmic factors of size $\log(1/\epsilon)$.*

Proof. This follows directly from (51) and (52), and then using $k_0 = O(\log(1/\epsilon))$ from Lemma 3.10. ∎

4. Projected Proximal Gradient Subproblem Solver

In this section we introduce our new solver for the nonsmooth trust-region subproblem (c.f. (13))

$$p^* \approx \operatorname*{argmin}_{s \in \mathbb{R}^d} m(p) := c + g^\top p + \frac{1}{2} p^\top H p + h(x + p), \quad \text{s.t.} \quad \|p\| \leq \Delta, \tag{63}$$

where here $x \in \mathbb{R}^d$ is the current iterate, $c \in \mathbb{R}$, $g \in \mathbb{R}^d$ and $H \in \mathbb{R}^{d \times d}$ (symmetric) form the current quadratic approximation for f and $\Delta > 0$ is the current trust-region radius. Our approach is based on minimizing m using the proximal gradient method [Bec17, Chapter 10.2], with the constraint $\|p\| \leq \Delta$ enforced at the end by projecting into the feasible region.

A full specification is given in Algorithm 2. In particular, we note that the stepsize γ for the proximal gradient part is a (to-be-specified) input, and to reduce the total effort we terminate the proximal gradient part early if we move too far outside the trust-region, $\|u_i - x\| > \mu_u \Delta_k$ (for $\mu_u > 1$).

First, we lower bound the decrease in m achieved in the first iteration.

Lemma 4.1. *Suppose h satisfies Assumption 3.1(ii), $\|H\| \leq L$ and $\lambda := \lambda_{\min}(H)$. Then for any iteration i of Algorithm 2 we have*

$$m(u_i - x) - m(u_{i+1} - x) \geq \left(\frac{1}{\gamma} - L + \frac{\lambda}{2}\right) \|u_{i+1} - u_i\|^2. \tag{64}$$

Proof. From the iteration $u_{i+1} = \text{Prox}_{\gamma h}(u_i - \gamma g - \gamma H(u_i - x))$ and the optimality condition for the proximity operator (Proposition 2.2) we get

$$u_i - u_{i+1} - \gamma g - \gamma H(u_i - x) \in \gamma \partial h(u_{i+1}), \tag{65}$$

and so

$$\frac{1}{\gamma}(u_i - u_{i+1}) + H(u_{i+1} - u_i) \in g + H(u_{i+1} - x) + \partial h(u_{i+1}) = \partial m(u_{i+1} - x). \tag{66}$$

By assumption, m is λ-convex (see Example 2.4) and so from Proposition 2.5 we have

$$\begin{aligned} m(u_i - x) - m(u_{i+1} - x) &\geq \left\langle \frac{1}{\gamma}(u_i - u_{i+1}) + H(u_{i+1} - u_i), u_i - u_{i+1} \right\rangle \\ &\qquad + \frac{\lambda}{2}\|(u_i - x) - (u_{i+1} - x)\|^2 && (67) \\ &= \frac{1}{\gamma}\|u_{i+1} - u_i\|^2 - (u_{i+1} - u_i)^\top H(u_{i+1} - u_i) + \frac{\lambda}{2}\|u_{i+1} - u_i\|^2 && (68) \\ &\geq \left(\frac{1}{\gamma} - L + \frac{\lambda}{2}\right)\|u_{i+1} - u_i\|^2, && (69) \end{aligned}$$

which completes the proof. ■

We can now show that Algorithm 2 achieves sufficient decrease in the model.

Theorem 4.2. *Suppose h satisfies Assumption 3.1(ii), $\|H\| \leq L$ and $\lambda := \lambda_{\min}(H)$. Then the output p^* of Algorithm 2 satisfies*

$$m(0) - m(p^*) \geq \theta \|u_1 - x\| \min(\Delta, \|u_1 - x\|), \tag{70}$$

where

$$\theta := \frac{1}{\gamma} - L + \frac{\lambda}{2} + \min\left(0, \frac{\lambda R(\gamma L)^2}{2}\right), \tag{71}$$

and where for any $t > 0$ we define $R(t) := \sum_{i=0}^{N-1}(1+t)^i = \frac{(1+t)^N - 1}{t} > 1$.

Proof. Suppose the loop in Algorithm 2 terminates after $n \in \{1, \ldots, N\}$ iterations (note that the termination conditions ensure at least one loop iteration is run).

Since $\mathrm{Prox}_{\gamma h}$ is nonexpansive (Proposition 2.2), for any $i \in \{0, \ldots, n-2\}$,

$$\|u_{i+2} - u_{i+1}\| \leq \|u_{i+1} - u_i - \gamma H(u_{i+1} - u_i)\| \leq (1 + \gamma L)\|u_{i+1} - u_i\|, \tag{72}$$

and so by induction we have $\|u_{i+1} - u_i\| \leq (1+\gamma L)^i \|u_1 - u_0\|$ for all $i \in \{0, \ldots, n-1\}$. By definition of $R(t)$, this gives us

$$\|u_n - x\| = \|u_n - u_0\| \leq \sum_{i=0}^{n-1} \|u_{i+1} - u_i\| \leq R(\gamma L)\|u_1 - u_0\| = R(\gamma L)\|u_1 - x\|, \tag{73}$$

where the second inequality uses $n \leq N$. Next, we apply Lemma 4.1 to obtain

$$\begin{aligned} m(0) - m(u_n - x) &= \sum_{i=0}^{n-1} m(u_i - x) - m(u_{i+1} - x), && (74) \\ &\geq \left(\frac{1}{\gamma} - L + \frac{\lambda}{2}\right) \sum_{i=0}^{n-1} \|u_{i+1} - u_i\|^2, && (75) \\ &\geq \left(\frac{1}{\gamma} - L + \frac{\lambda}{2}\right) \|u_1 - x\|^2, && (76) \end{aligned}$$

where the last line uses $n \geq 1$, yielding $\sum_{i=0}^{n-1} \|u_{i+1} - u_i\|^2 \geq \|u_1 - u_0\|^2 = \|u_1 - x\|^2$.

Now, define $\alpha := \frac{\Delta}{\max(\Delta, \|u_n - x\|)} \in (0, 1]$, so the output of Algorithm 2 is $p^* = \alpha(u_n - x)$. Writing $p^* = (1-\alpha)0 + \alpha(u_n - x)$ and using the λ-convexity of m (Proposition 2.5) we get

$$(1-\alpha)m(0) + \alpha m(u_n - x) \geq m(p^*) + \frac{\lambda}{2}\alpha(1-\alpha)\|u_n - x\|^2, \tag{77}$$

and so from (76) we have

$$m(0) - m(p^*) \geq \alpha(m(0) - m(u_n - x)) + \frac{\lambda}{2}\alpha(1-\alpha)\|u_n - x\|^2, \tag{78}$$

$$\geq \alpha\left(\frac{1}{\gamma} - L + \frac{\lambda}{2}\right)\|u_1 - x\|^2 + \frac{\lambda}{2}\alpha(1-\alpha)\|u_n - x\|^2. \tag{79}$$

If $\lambda \geq 0$ then we use $\alpha \leq 1$ to conclude $\frac{\lambda(1-\alpha)}{2} \geq 0$, and if $\lambda < 0$ then we use $\alpha > 0$ to get $\frac{\lambda(1-\alpha)}{2} > \frac{\lambda}{2}$. Hence

$$\frac{\lambda(1-\alpha)}{2} \geq \min\left(0, \frac{\lambda}{2}\right), \tag{80}$$

and so

$$m(0) - m(p^*) \geq \alpha\left(\frac{1}{\gamma} - L + \frac{\lambda}{2}\right)\|u_1 - x\|^2 + \alpha\min\left(0, \frac{\lambda}{2}\right)\|u_n - x\|^2. \tag{81}$$

We then apply (73) to get

$$m(0) - m(p^*) \geq \alpha\left(\frac{1}{\gamma} - L + \frac{\lambda}{2}\right)\|u_1 - x\|^2 + \alpha\min\left(0, \frac{\lambda}{2}\right)R(\gamma L)^2\|u_1 - x\|^2 = \alpha\theta\|u_1 - x\|^2. \tag{82}$$

Lastly, if $\|u_1 - x\| \leq \Delta$ then $\alpha = 1$, so $\alpha\|u_1 - x\| = \|u_1 - x\| = \min(\Delta, \|u_1 - x\|)$. Instead, if $\|u_1 - x\| > \Delta$ then $\alpha = \frac{\Delta}{\|u_1 - x\|}$ and so $\alpha\|u_1 - x\| = \Delta = \min(\Delta, \|u_1 - x\|)$. In either case, we get the desired result. ∎

From Lemma 4.2, using the fact that $\lim_{t\to 0} R(t) = N$, we note that $\theta > 0$ holds provided we pick the stepsize $\gamma > 0$ to be sufficiently small (for any choice of total iterations N). However, in light of Assumption 3.2, the value of L may grow unboundedly over the iterations of the main trust-region algorithm (Algorithm 1). The below result shows that regardless of the value of L, there is a range of values of γ that achieve sufficient decrease.

Lemma 4.3. *Assume $\|H\| \leq L$ and $g = \nabla f(x)$, and we run Algorithm 2 with at most $N \geq 1$ iterations with stepsize $\gamma > 0$. Then there exists $c^* > 0$ and $\kappa_s > 0$ such that, for all $\gamma \in [c^*/(10L), c^*/L]$, we have*

$$m(0) - m(p^*) \geq \kappa_s \pi(x, \gamma)\min\left(\Delta, \frac{\pi(x,\gamma)}{L}\right). \tag{83}$$

The constants c^* and κ_s depend on N, but not γ or μ_u.

Proof. From Lemma 4.2 with the conservative bound $\lambda = -L$ and $\|u_1 - x\| = \gamma\pi(x,\gamma)$ by the definition of $\pi(x,\gamma)$ (6), we have

$$m(0) - m(p^*) \geq \left(1 - \frac{3\gamma L}{2} - \frac{\gamma L R(\gamma L)^2}{2}\right)\pi(x,\gamma)\min\left(\Delta, (\gamma L)\frac{\pi(x,\gamma)}{L}\right). \tag{84}$$

Now suppose that $\gamma = c/L$ for some $0 < c < 1$. Then

$$m(0) - m(p^*) \geq \left(1 - \frac{3c}{2} - \frac{cR(c)^2}{2}\right)\pi(x,\gamma)\min(1,c)\min\left(\Delta, \frac{\pi(x,\gamma)}{L}\right) \tag{85}$$

$$= c\left(1 - \frac{3c}{2} - \frac{cR(c)^2}{2}\right)\pi(x,\gamma)\min\left(\Delta, \frac{\pi(x,\gamma)}{L}\right) \tag{86}$$

Furthermore, since $R(c) = \sum_{i=0}^{N-1}(1+c)^i \leq N(1+c)^{N-1}$, a sufficient condition for the result to hold is therefore to choose κ_s such that

$$c\left(1 - \frac{3c}{2} - \frac{c[N(1+c)^{N-1}]^2}{2}\right) \geq \kappa_s > 0. \tag{87}$$

Since the factor inside the brackets approaches 1 as $c \to 0^+$, there exists c^* such that

$$1 - \frac{3c}{2} - \frac{c[N(1+c)^{N-1}]^2}{2} \geq \frac{1}{2}, \qquad \forall\, c \in (0, c^*]. \tag{88}$$

Thus (87) and hence the final result holds with, for example, $\kappa_s = \frac{c^*}{20}$ and any $c \in [c^*/10, c^*]$. ■

Our final result confirms that the sufficient decrease requirement Assumption 3.3 can be achieved by running Algorithm 2 with decreasing choices of γ, regardless of the size of $\|H\|$.

Corollary 4.4. *Suppose the assumptions of Lemma 4.3 hold. If we run Algorithm 2 with stepsize choices $\gamma_j = \alpha^j\gamma_0$ for some $\alpha \in (0.1, 1)$ and $j = 0, 1, 2, \dots,$ with $\gamma_0 \leq \gamma_{\max}$, then there is a finite j (possibly dependent on L and N) such that*

$$m(0) - m(p^*) \geq \kappa_s \pi(x, \gamma_{\max})\min\left(\Delta, \frac{\pi(x,\gamma_{\max})}{L}\right), \tag{89}$$

for some $\kappa_s > 0$ depending on N but independent of L.

Proof. Since $\alpha \in (0.1, 1)$, there must be at least one j for which $\gamma_j \in [c^*/(10L), c^*/L]$, and so Lemma 4.3 holds. The result follows from $\pi(x,\gamma_j) \geq \pi(x,\gamma_0) \geq \pi(x,\gamma_{\max})$ (Lemma 2.3 with $\gamma_j \leq \gamma_0 \leq \gamma_{\max}$). ■

Remark 4.5. In our numerical experiments, follow the approach in Corollary 4.4, where we accept the current choice of γ_j if all computed iterates u_i of Algorithm 2 satisfy $m(0) - m(u_i) > 0$ and $m(0) - m(p^*) > 0$. For the backtracking, we use the heuristic choice $\gamma_0 = \frac{2\|g_0\|}{3\|H_0 g_0\|}$, where g_0 and H_0 are the model gradient and Hessian at the first iteration. This choice arises from setting $\frac{1}{\gamma_0} - L + \frac{\lambda}{2} = 0$ (compare with Lemma 4.1) under the conservative assumption $\lambda = -L$, and where $L = \|H_0\|$ is approximated by one iteration of the power method, $\|H_0\| \geq \|H_0 g_0\|/\|g_0\|$. The successful value γ_j in one iteration of Algorithm 1 is then taken to be the value of γ_0 in the next iteration. We do not change the value of N across iterations of Algorithm 1.

5. Numerical Experiments

In our experiments we compare Algorithm 1 with different choices of subproblem solver. Our implemented version of Algorithm 1 uses $\Delta_0 = 1$ and a slightly more sophisticated mechanism for choosing x_{k+1} and Δ_{k+1}, similar to [NW06, Algorithm 4.1]. Specifically, we set $x_{k+1} = x_k + p_k$ if $\rho_k \geq 10^{-3}$ and $x_{k+1} = x_k$ otherwise, and

$$\Delta_{k+1} = \begin{cases} \min(2\Delta_k, 10^{10}), & \rho_k \geq 0.75 \text{ and } \|p_k\| \geq (1-10^{-5})\Delta_k, \\ 0.5\Delta_k, & \rho_k < 0.25, \\ \Delta_k, & \text{otherwise.} \end{cases} \tag{90}$$

This modification does not affect the complexity theory developed in Section 3, and was chosen for practicality. The two choices of subproblem solver we compare for this version of Algorithm 1 are:

- PPG: the projected proximal gradient method developed in Algorithm 2; and
- SPG: the spectral proximal gradient method given in [BK23a, Algorithm 3].

The SPG method was chosen for comparison as it was the top-performing method of the 5 subproblem solvers compared in [BK23a] (excluding one method that specifically exploits an L^1 regularization structure rather than just using $\text{Prox}_{\gamma h}$). Both methods use comparable problem information: they interact with the model Hessian H_k only through Hessian-vector products, and with the nonsmooth term h through only evaluations of h and its proximity operator. For PPG we use $\mu_u = 2$ and $\alpha = 0.9$ (α is used for backtracking; see Corollary 4.4 and Remark 4.5) and for SPG we set $t_{\min} = 10^{-12}$, $t_{\max} = 10^{12}$, $\overline{\tau} = 10^{-5}$, $t_0 = 1$ and $\tau_k = 10^{-3}h_k$ (with all values except t_0 taken from [BK23a]). We consider the impact of varying the maximum iterations N for both PPG and SPG subproblem solvers, and show results for $N \in \{15, 30, 50\}$, where $N = 15$ is the value used in [BK23a].

For our test problems (1), we use $h(x) = \|x\|_1$ and take f to be from a collection of 154 unconstrained CUTEst problems [GOT15, FRB22] with dimension $d \in [2, 50]$ based on [Rag22, Appendix A.1]. These problems are listed in Appendix A. For all problems we use the exact Hessian $H_k = \nabla^2 f(x_k)$ to build our model (12). We run all test problems until we reach first-order optimality level $\pi_k(1) = \pi(x_k, 1) \leq 10^{-6}$, capped at 10^4 iterations of Algorithm 1.

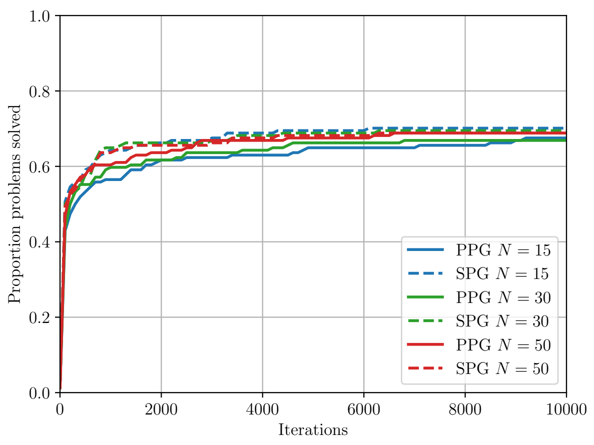
To compare our (subproblem) solvers, we measure the number of iterations taken to achieve $\pi_k(1) \leq \tau$ for the first time, for some accuracy level $\tau \ll 1$. We plot both data [MW09] and performance profiles [DM02]. For solvers $S \in \mathcal{S}$ and problems $P \in \mathcal{P}$, let $K(S, P, \tau)$ be the first iteration with $\pi_k(1) \leq \tau$ (with $K = +\infty$ if this never occurs). For a given solver S and accuracy level τ, data profiles measure the proportion of problems solved after some fixed number of iterations,

$$d_{S,\tau}(\alpha) = \frac{1}{|\mathcal{P}|}|\{P : K(S, P, \tau) \leq \alpha\}|, \qquad \forall \alpha \geq 0, \tag{91}$$

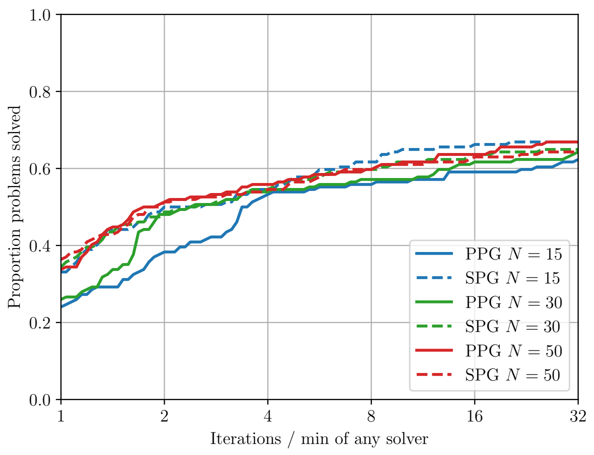
and performance profiles measure the proportion of problems solved within some ratio of the fastest solver for that problem,

$$p_{S,\tau}(\alpha) = \frac{1}{|\mathcal{P}|}|\{P : K(S, P, \tau) \leq \alpha \min_{S' \in \mathcal{S}} K(S', P, \tau)\}|, \qquad \forall \alpha \geq 1. \tag{92}$$

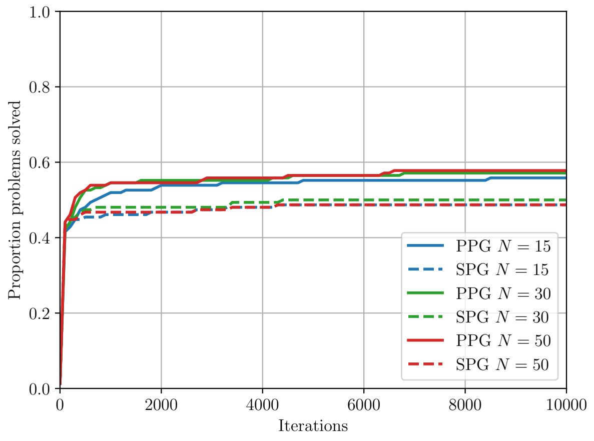
Our results compare PPG and SPG subproblem solvers with $N \in \{15, 30, 50\}$ for accuracy levels $\tau \in \{10^{-3}, 10^{-6}\}$, with data and performance profiles given in Figure 1.



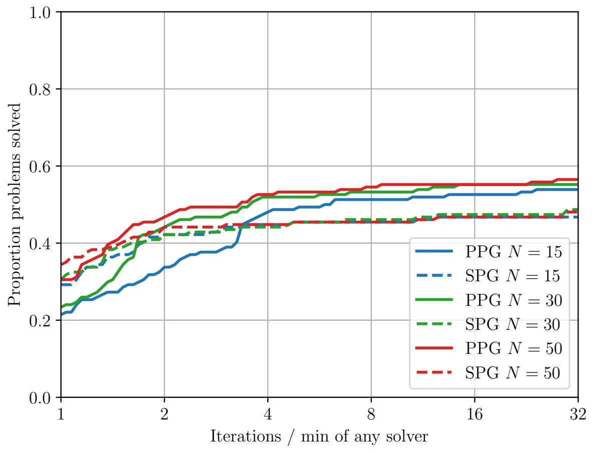
(a) Data profile, $\tau = 10^{-3}$



(b) Performance profile, $\tau = 10^{-3}$



(c) Data profile, $\tau = 10^{-6}$



(d) Performance profile, $\tau = 10^{-6}$

Figure 1. Comparison of PPG (Algorithm 2) and SPG [BK23a] subproblem solvers with increasing maximum iterations $N \in \{15, 30, 50\}$ *in the trust-region method Algorithm 1.*

For low accuracy solutions $\tau = 10^{-3}$, the three SPG subproblem solvers outperform the PPG subproblem solvers, with the PPG solvers improving in performance with larger values of N. The best PPG variant, $N = 50$, has comparable performance to the SPG solvers.

However, for high accuracy solutions $\tau = 10^{-6}$, all the PPG variants notably outperform all the SPG variants. We also note that increasing the value of N for PPG continues to improve the performance, where increasing N for SPG has limited benefit.

Thus, our results suggest that PPG with larger N is a more effective subproblem solver than PPG with smaller N, and that PPG can outperform SPG when higher accuracy solutions are desired.

6. Conclusion and Future Work

We extended the theoretical analysis of the nonsmooth trust-region method from [BK23b] to prove worst-case complexity bounds in the case of unbounded model Hessians, and showed results that match the smooth case [DHO24]. We also introduced the projected proximal gradient nonsmooth trust-region subproblem solver, a simple method that demonstrates good numerical performance, particularly when high-accuracy solutions of the main problem (1) are desired. Potential directions for future work include extending our results to the derivative-free case (i.e. where ∇f) is not available, and handling more complicated h via inexact proximity operator evaluations.

Acknowledgments

This research was initiated during HMP's visit to the Sydney Mathematical Research Institute (SMRI), Australia in May 2024 and HMP and MND's visit to the Vietnam Institute for Advanced Study in Mathematics (VIASM), Vietnam in June 2024, for which HMP and MND are grateful for the generous support and hospitality of SMRI and VIASM. MND is partially supported by the Australian Research Council Discovery Project DP230101749. LR is supported by the Australian Research Council Discovery Early Career Award DE240100006.

References

[ABO22a] Aleksandr Y. Aravkin, Robert Baraldi, and Dominique Orban. A Levenberg-Marquardt method for nonsmooth regularized least squares. arXiv preprint arXiv:2301.02347, 2022.

[ABO22b] Aleksandr Y. Aravkin, Robert Baraldi, and Dominique Orban. A proximal quasi-Newton trust-region method for nonsmooth regularized optimization. *SIAM Journal on Optimization*, 32(2):900–929, 2022.

[BE19] James V. Burke and Abraham Engle. Line search and trust-region methods for convex-composite optimization. arXiv preprint arXiv:1806.05218, 2019.

[BE20] James V. Burke and Abraham Engle. Strong metric (sub)regularity of Karush–Kuhn–Tucker mappings for piecewise linear-quadratic convex-composite optimization and the quadratic convergence of Newton's method. *Mathematics of Operations Research*, 45(3):1164–1192, 2020.

[Bec17] Amir Beck. *First-order methods in optimization.* SIAM, 2017.

[BK23a] Robert J Baraldi and Drew P Kouri. Efficient proximal subproblem solvers for a nonsmooth trust-region method. Optimization Online, https://optimization-online.org/?p=24879, 2023.

[BK23b] Robert J. Baraldi and Drew P. Kouri. A proximal trust-region method for nonsmooth optimization with inexact function and gradient evaluations. *Mathematical Programming*, 201(1-2):559–598, 2023.

[BK24] Robert J. Baraldi and Drew P. Kouri. Local convergence analysis of an inexact trust-region method for nonsmooth optimization. *Optimization Letters*, 18(3):663–680, 2024.

[CGST93] A. R. Conn, Nick Gould, A. Sartenaer, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints. *SIAM Journal on Optimization*, 3(1):164–221, 1993.

[CGT88] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25(2):433–460, 1988.

[CGT00] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-Region Methods*, volume 1 of *MPS-SIAM Series on Optimization.* MPS/SIAM, Philadelphia, 2000.

[CGT12] C. Cartis, N. I. M. Gould, and P. L. Toint. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. *IMA Journal of Numerical Analysis*, 32(4):1662–1695, 2012.

[CGT22] Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives.* Number 30 in MOS-SIAM Series on Optimization. MOS/SIAM, Philadelphia, 2022.

[CMW23] Ziang Chen, Andre Milzarek, and Zaiwen Wen. A trust-region method for nonsmooth nonconvex optimization. *Journal of Computational Mathematics*, 41(4):639–672, 2023.

[CNY13] Xiaojun Chen, Lingfeng Niu, and Yaxiang Yuan. Optimality conditions and a smoothing trust region Newton method for nonlipschitz optimization. *SIAM Journal on Optimization*, 23(3):1528–1552, 2013.

[DHO24] Youssef Diouane, Mohamed Laghdaf Habiboullah, and Dominique Orban. Complexity of trust-region methods in the presence of unbounded Hessian approximations. arXiv preprint arXiv:2408.06243, 2024.

[DLT95] J. E. Dennis, Jr., Shou-Bai B Li, and Richard A Tapia. A unified approach to global convergence of trust region methods for nonsmooth optimization. *Mathematical Programming*, 68(1-3):319–346, 1995.

[DM02] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[DP19] M. N. Dao and H. M. Phan. Adaptive Douglas–Rachford splitting algorithm for the sum of two operators. *SIAM Journal on Optimization*, 29(4):2697–2724, 2019.

[DPT24] M. N. Dao, T. N. Pham, and P. T. Tung. Doubly relaxed forward-Douglas–Rachford splitting for the sum of two nonconvex and a DC function. arXiv preprint arXiv:2405.08485, 2024.

[ER21] Matthias J Ehrhardt and Lindon Roberts. Inexact derivative-free optimization for bilevel learning. *Journal of Mathematical Imaging and Vision*, 63(5):580–600, 2021.

[FRB22] Jaroslav Fowkes, Lindon Roberts, and Árpád Bűrmen. PyCUTEst: An open source Python package of optimization test problems. *Journal of Open Source Software*, 7(78):4377, 2022.

[GJV16] R. Garmanjani, D. Júdice, and L. N. Vicente. Trust-region methods without using derivatives: Worst case complexity and the nonsmooth case. *SIAM Journal on Optimization*, 26(4):1987–2011, 2016.

[GOT15] Nicholas I. M. Gould, Dominique Orban, and Philippe L. Toint. CUTEst: A constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557, 2015.

[GYY16] Geovani Nunes Grapiglia, Jinyun Yuan, and Ya-xiang Yuan. A derivative-free trust-region algorithm for composite nonsmooth optimization. *Computational and Applied Mathematics*, 35(2):475–499, 2016.

[HR22] Matthew Hough and Lindon Roberts. Model-based derivative-free methods for convex-constrained optimization. *SIAM Journal on Optimization*, 32(4):2552–2579, 2022.

[HSW12] Roland Herzog, Georg Stadler, and Gerd Wachsmuth. Directional sparsity in optimal control of partial differential equations. *SIAM Journal on Control and Optimization*, 50(2):943–963, 2012.

[KL21] Christian Kanzow and Theresa Lechner. Globalized inexact proximal Newton-type methods for nonconvex composite functions. *Computational Optimization and Applications*, 78(2):377–410, 2021.

[KSD10] Dongmin Kim, Suvrit Sra, and Inderjit Dhillon. A scalable trust-region algorithm with application to mixed-norm regression. In *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

[LLR24] Yanjun Liu, Kevin H. Lam, and Lindon Roberts. Black-box optimization algorithms for regularized least-squares problems. arXiv preprint arXiv:2407.14915, 2024.

[LO23] Geoffroy Leconte and Dominique Orban. The indefinite proximal gradient method. arXiv preprint arXiv:2309.08433, 2023.

[LSS14] Jason D. Lee, Yuekai Sun, and Michael A. Saunders. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.

[LW19] Ching-pei Lee and Stephen J. Wright. Inexact successive quadratic approximation for regularized optimization. *Computational Optimization and Applications*, 72(3):641–674, 2019.

[Mit70] D. S. Mitrinović. *Analytic Inequalities.* Springer-Verlag, 1970.

[Mor06] B. S. Mordukhovich. *Variational Analysis and Generalized Differentiation I. Basic Theory.* Springer, 2006.

[MW09] Jorge J. Moré and Stefan M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[NW06] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[OM24] Wenqing Ouyang and Andre Milzarek. A trust region-type normal map-based semismooth Newton method for nonsmooth nonconvex composite optimization. *Mathematical Programming*, 2024.

[Pow84] M. J. D. Powell. On the global convergence of trust region algorithms for unconstrained minimization. *Mathematical Programming*, 29:297–303, 1984.

[QS94] Liqun Qi and Jie Sun. A trust region algorithm for minimization of locally Lipschitzian functions. *Mathematical Programming*, 66(1-3):25–43, 1994.

[Rag22] Tom Maël Ragonneau. *Model-Based Derivative-Free Optimization Methods and Software*. PhD thesis, Hong Kong Polytechnic University, 2022.

[Toi88] Philippe L. Toint. Global convergence of a a of trust-region methods for nonconvex minimization in Hilbert space. *IMA Journal of Numerical Analysis*, 8(2):231–252, 1988.

[WR22] Stephen J. Wright and Benjamin Recht. *Optimization for Data Analysis*. Cambridge University Press, 2022.

[ZYZ24] Shimin Zhao, Tao Yan, and Yuanguo Zhu. Proximal gradient algorithm with trust region scheme on Riemannian manifold. *Journal of Global Optimization*, 88(4):1051–1076, 2024.

A. List of Test Problems

The following table contains a list of the 154 CUTEst problems [GOT15, FRB22] used for the numerical experiments in Section 5 and their dimension $d \in [2, 50]$, based on the collection from [Rag22, Appendix A.1]. Any values in brackets after the problem name are the optional problem parameters used.

Name (params)	d	Name (params)	d	Name (params)	d
AKIVA	2	FREUROTH ($N = 10$)	10	PALMER2C	8
ALLINITU	4	GAUSSIAN	3	PALMER3C	8
ARGLINA ($N = 10$)	10	GBRAINLS	2	PALMER4C	8
ARGLINB ($N = 10$)	10	GENHUMPS ($N = 5$)	5	PALMER5C	6
ARGLINC ($N = 10$)	10	GENROSE ($N = 5$)	5	PALMER5D	4
ARGTRIGLS ($N = 10$)	10	GROWTHLS	3	PALMER6C	8
BARD	3	GULF	3	PALMER7C	8
BEALE	2	HAHN1LS	7	PALMER8C	8
BENNETT5LS	3	HAIRY	2	PENALTY1 ($N = 10$)	10
BIGGS6	6	HATFLDD	3	PENALTY2 ($N = 10$)	10
BOX ($N = 10$)	10	HATFLDE	3	POWELLBSLS	2
BOX3	3	HATFLDFL	3	POWELLSG ($N = 16$)	16
BOXBODLS	2	HEART6LS	6	POWER ($N = 20$)	20
BOXPOWER ($N = 10$)	10	HEART8LS	8	QUARTC ($N = 25$)	25
BROWNAL ($N = 10$)	10	HELIX	3	RAT42LS	3
BROWNBS	2	HIELOW	3	ROSENBR	2
BROWNDEN	4	HILBERTA ($N = 5$)	5	ROSENBRTU	2
BROYDN3DLS ($N = 10$)	10	HILBERTB ($N = 10$)	10	ROSZMAN1LS	4
BROYDNBDLS ($N = 10$)	10	HIMMELBB	2	S308	2
BRYBND ($N = 10$)	10	HIMMELBF	4	SBRYBND ($N = 10$)	10
CHNROSNB ($N = 25$)	25	HIMMELBG	2	SCHMVETT ($N = 3$)	3
CHNRSNBM ($N = 25$)	25	HIMMELBH	2	SCOSINE ($N = 10$)	10
CHWIRUT1LS	3	HUMPS	2	SCURLY10 ($N = 10$)	10
CHWIRUT2LS	3	INDEFM ($N = 10$)	10	SENSORS ($N = 3$)	3
CLIFF	2	JENSMP	2	SINEVAL	2
COSINE ($N = 10$)	10	KIRBY2LS	5	SINQUAD ($N = 5$)	5
CUBE	2	KOWOSB	4	SISSER	2
DENSCHNA	2	LANCZOS1LS	6	SNAIL	2
DENSCHNB	2	LANCZOS2LS	6	SPARSINE ($N = 10$)	10
DENSCHNC	2	LANCZOS3LS	6	SPARSQUR ($N = 10$)	10
DENSCHND	3	LIARWHD ($N = 36$)	36	SSBRYBND ($N = 10$)	10
DENSCHNE	3	LOGHAIRY	2	SSCOSINE ($N = 10$)	10
DENSCHNF	2	LSC1LS	3	SSI	3
DIXON3DQ ($N = 10$)	10	LSC2LS	3	STREG	4
DJTL	2	MANCINO ($N = 20$)	20	THURBERLS	7
DQDRTIC ($N = 10$)	10	MARATOSB	2	TOINTGOR	50
DQRTIC ($N = 10$)	10	MEXHAT	2	TOINTGSS ($N = 10$)	10
ECKERLE4LS	3	MEYER3	3	TOINTPSP	50
EDENSCH ($N = 36$)	36	MGH09LS	4	TOINTQOR	50
ENGVAL1 ($N = 2$)	2	MGH10LS	3	TQUARTIC ($N = 10$)	10
ENGVAL2	3	MISRA1BLS	2	TRIDIA ($N = 20$)	20
ENSOLS	9	MISRA1DLS	2	VARDIM ($N = 10$)	10
ERRINROS ($N = 25$)	25	MOREBV ($N = 10$)	10	VAREIGVL ($N = 19$)	20
ERRINRSM ($N = 25$)	25	NCB20B ($N = 22$)	22	VESUVIALS	8
EXPFIT	2	NONCVXU2 ($N = 10$)	10	VESUVIOLS	8
EXTROSNB ($N = 5$)	5	NONCVXUN ($N = 10$)	10	VESUVIOULS	8
FBRAIN3LS	6	NONDIA ($N = 20$)	20	VIBRBEAM	8
FLETBV3M ($N = 10$)	10	OSBORNEB	11	WATSON ($N = 12$)	12
FLETCBV2 ($N = 10$)	10	OSCIGRAD ($N = 10$)	10	YFITU	3
FLETCBV3 ($N = 10$)	10	OSCIPATH ($N = 5$)	5	ZANGWIL2	2
FLETCHBV ($N = 10$)	10	PALMER1C	8		
FLETCHCR ($N = 10$)	10	PALMER1D	7		

Table 1. List of CUTEst problems used for numerical experiments.