Globalized Inexact Proximal Newton-type Methods for Nonconvex Composite Functions

Christian Kanzow* Theresa Lechner*

February 27, 2020

Abstract

Optimization problems with composite functions consist of an objective function which is the sum of a smooth and a (convex) nonsmooth term. This particular structure is exploited by the class of proximal gradient methods and some of their generalizations like proximal Newton and quasi-Newton methods. The current literature on these classes of methods almost exclusively considers the case where also the smooth term is convex. Here we present a globalized proximal Newton-type method which allows the smooth term to be nonconvex. The method is shown to have nice global and local convergence properties, and some numerical results indicate that this method is very promising also from a practical point of view.

1 Introduction

In this paper we deal with the composite problem

$$\min_{x\in\mathbb{R}^n} \psi(x) := f(x) + \varphi(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is (twice) continuously differentiable and $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is convex, proper, and lower semicontinuous (lsc). In this formulation, the objective function ψ is neither convex nor smooth, so it covers a wide class of problems in statistics, machine learning, compressed sensing, and signal processing. Since φ is allowed to take the value $+\infty$, (1) also covers constrained problems on convex sets.

1.1 Background

Optimization problems in the form (1) arise in many applications, cf. the list given by Combettes and Ways [11] and references therein. The function f often represents a smooth loss function such as the quadratic loss $f(x) := \|Ax - b\|_2^2$ or the logistic loss

*University of Würzburg, Institute of Mathematics, Emil-Fischer-Str. 30, 97074 Würzburg, Germany; kanzow@mathematik.uni-wuerzburg.de, theresa.lechner2@mathematik.uni-wuerzburg.de

$f(x) := \frac{1}{m}\sum_{i=1}^{m} \log\left(1+\exp(a_i^T x)\right)$ for some given data $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$, and $a_i \in \mathbb{R}^n$ for $i = 1, \ldots, m$. The basic aim is then to find a minimizer of f, e.g. for image deblurring or to classify some data. A convex regularizer φ is added to cover some additional constraints or to control some sparsity. Typical regularizers are the ℓ_1- and ℓ_2-norm, a weighted ℓ_1-norm $\varphi(x) := \sum_{i=1}^{n} \omega_i |x_i|$ for some weights $\omega_i > 0$, or the total variation $\varphi(x) = \|\nabla x\| := \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

1.2 Description of the Method

In every step of the proximal Newton-type method, we (inexactly) solve the problem

$$\arg\min_y \left\{ f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T H (y-x) + \varphi(y) \right\} \tag{2}$$

for some $x \in \mathbb{R}^n$ and a given matrix H which is either equal to the Hessian $\nabla^2 f(x)$ or represents a suitable approximation of the exact Hessian. The advantage of using proximal Newton-type steps that take into account second order information of f is that, similar to smooth Newton-type methods, one can prove fast local convergence. However, they are only well-defined for convex f and the convergence theorems typically require some strong convexity assumption.

In contrast, proximal gradient methods perform a backward step using only first order information of f. This means that (2) is solved for some positive definite $H \in \mathbb{R}^{n\times n}$, which is usually a fixed multiple of the identity matrix. The method can therefore be shown to converge globally in the sense that every accumulation point of a sequence generated by this method is a stationary point of ψ, but it is not possible to achieve fast local convergence results.

In this paper, we take into account the advantages of both methods and join them to get a globalized proximal Newton-type method. Since the proximal Newton-type update is preferable, we try to solve the corresponding subproblem and use a novel descent condition to control whether the current iterate is updated with its solution or a proximal gradient step is performed. To achieve global convergence, we further add an Armijo-type line search.

As the solution of the steps requires a high amount of computing, our convergence theory allows some freedom in the choice of the matrices H, in particular, one can use quasi-Newton or limited memory quasi-Newton matrices.

1.3 Related Work

The original proximal gradient method was introduced by Fukushima and Mine [16]. It may be viewed as a special instance of the method described in Tseng and Yun [37], which utilizes a block separable structure of φ and performs block wise descent. Numerous authors [18, 29, 38] deal with acceleration techniques whereby all of them require the Lipschitz continuity of the gradient ∇f. Further methods [5, 33] also assume that f is convex.

In an intermediate approach between proximal Newton and proximal gradient methods, the matrix H in (2) does not need to be a multiple of the identity matrix, but is still positive definite, uniformly bounded, and does not necessarily contain second order information of f. Various line search techniques and inexactness conditions on the subproblem solution can be applied [7, 15, 17, 20, 21, 34, 35] to prove global convergence. These references include fast local convergence results for the case that H is replaced by the Hessian of f or some approximation and a suitable boundedness condition holds.

In Lee, Sun, and Saunders [21] a generic version of the proximal Newton method is presented and several convergence results based on the exactness of the subproblem solutions and the Hessian approximation are stated. For the local convergence theory, they need strong convexity of f. In Yue, Zhou, and So [39], an inexact proximal Newton method with regularized Hessian is presented which assumes f to be convex, but not strongly convex, and an error bound condition. Their inexactness criterion is similar to ours. The authors in [22, 36] assume that f is convex and self-concordant and apply a damped proximal Newton method.

Further methods exist for the case where we can write $\varphi = \tilde{\varphi} \circ B$ for a linear mapping $B\colon \mathbb{R}^n \to \mathbb{R}^p$ and a convex function $\tilde{\varphi}\colon \mathbb{R}^p \to \mathbb{R}$. This formulation is used if the proximity operator of $\tilde{\varphi}$ is easy to compute whereas the one of φ is not. In [9, 10, 23] fixed point methods are used to solve the problems under different assumptions, the reformulation into a constrained problem is applied in [2, 40].

Another class of methods to solve (1) are semismooth Newton methods. Patrinos, Stella and Bemporad assume in [31] that f is convex and apply a semismooth Newton method combined with a line search strategy. For strongly convex f with Lipschitz continuous gradient, Patrinos and Bemporad [30] state a semismooth Newton method that uses a globalization strategy similar to our method and applies a proximal gradient step if the given descent criterion does not hold. A semismooth Newton method with filter globalization is introduced by Milzarek and Ulbrich [26] for $\varphi(x) = \lambda\|x\|_1$ with some $\lambda > 0$ and adapted for arbitrary convex φ by Milzarek [25]. For the semismooth update, they check a filter condition and, if it does not hold, a proximal gradient step with Armijo-type line search is performed.

1.4 Outline of the paper

This paper is organized as follows. First, we introduce the proximity operator with some properties, formulate the proximal gradient method, and state a convergence result in Section 2. The globalization of the proximal Newton-type method and its inexact variant is deduced in Section 3, where we also state some preliminary observations. In Section 4, we first prove global convergence under fairly mild assumptions, and then provide a fast local convergence result. We then consider the numerical behaviour of our method(s) on different classes of problems in Section 5, also including a comparison with several state-of-the-art solvers. We conclude with some final remarks in Section 6.

1.5 Notation

For $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ and $J \subset \{1, \dots, n\}$, the subvector $x_J \in \mathbb{R}^{|J|}$ consists of all elements x_i of x with $i \in J$. Furthermore, $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ is the set of extended real numbers. The set of all symmetric matrices in $\mathbb{R}^{n \times n}$ is denoted by $\mathbb{S}^n$, and the set of all symmetric positive definite matrices is abbreviated by $\mathbb{S}^n_{++}$. We write $H \succ 0$ or $H \succeq 0$ for $H \in \mathbb{R}^{n \times n}$ if H is positive definite or positive semidefinite, respectively. Analogously, we write $H \succ G$ or $H \succeq G$ for $G, H \in \mathbb{R}^{n \times n}$ if $H - G$ is positive (semi)definite. Finally, we write $\|x\|_H := \sqrt{x^T H x}$ for the norm induced by a given matrix $H \succ 0$.

2 The Proximal Gradient Method

This section first recalls the definition and some elementary properties of the proximity operator, and then describes a version of the proximal gradient method which is applicable to possibly nonconvex composite optimization problems. Throughout this section, we assume that f is continuously differentiable and φ is proper, lsc, and convex.

2.1 The Proximity Operator

The proximity operator was introduced by Moreau [28] and turned out to be a very useful tool both from a theoretical and an algorithmic point of view. Here we restate only some of its properties, and refer to the monograph [3] by Bauschke and Combettes for more details.

For a positive definite matrix $H \in \mathbb{R}^{n \times n}$ and a convex, proper, and lsc function $\varphi : \mathbb{R}^n \to \overline{\mathbb{R}}$, the mapping

$$x \mapsto \operatorname{prox}_\varphi^H(x) := \arg\min_y \left\{ \varphi(y) + \frac{1}{2}\|y - x\|_H^2 \right\}$$

is called the *proximity operator* of φ with respect to H. Here, the minimizer $\operatorname{prox}_\varphi^H(x)$ is uniquely defined for all $x \in \mathbb{R}^n$ since the expression inside the arg min is a strongly convex function. If H is the identity matrix, we simply write $\operatorname{prox}_\varphi(x)$ instead of $\operatorname{prox}_\varphi^I(x)$.

Using Fermat's rule and the sum rule for subdifferentials, the definition of the proximity operator gives $p = \operatorname{prox}_\varphi^H(x)$ if and only if $0 \in \partial\varphi(p) + H(p - x)$, or equivalently

$$p \in x - H^{-1}\partial\varphi(p). \tag{3}$$

We next restate a result on the continuity of the proximity operator due to Milzarek [25, Corollary 3.1.4], which states that the proximity operator is continuous not only with respect to the argument, but also with respect to the positive definite matrix.

Lemma 2.1. *The proximity operator* $\operatorname{prox}_\varphi : \mathbb{R}^n \times \mathbb{S}^n, (x, H) \mapsto \operatorname{prox}_\varphi^H(x)$ *is Lipschitz continuous on every compact subset of* $\mathbb{R}^n \times \mathbb{S}^n_{++}$*, and continuous on* $\mathbb{R}^n \times \mathbb{S}^n_{++}$*.*

We call $x^* \in \operatorname{dom}\varphi$ a *stationary point* of the program (1) if $0 \in \nabla f(x^*) + \partial\varphi(x^*)$. Using [3, Proposition 17.14] and (3), we obtain the characterizations

$$\begin{aligned} x^* \text{ stationary point of (1)} &\iff -\nabla f(x^*) \in \partial\varphi(x^*) \\ &\iff \psi'(x^*; d) \geq 0 \text{ for all } d \in \mathbb{R}^n \\ &\iff x^* = \operatorname{prox}_{\varphi}^{H}\left(x^* - H^{-1}\nabla f(x^*)\right), \end{aligned}$$

where the last reformulation turns out to be independent of the particular matrix H.

2.2 Proximal Gradient Method

The proximal gradient method was introduced by Fukushima and Mine [16] as a generalization of the proximal point algorithm, which, in turn, was established by Rockafellar [32]. Note that the existing literature on the proximal gradient method usually assumes f to be both convex and smooth with a (globally) Lipschitz continuous gradient. The assumptions are required in order to obtain complexity and rate of convergence results, cf. Beck [4] for more details.

Here we present a version of the proximal gradient method which still has nice global convergence properties also in the case where f is only continuously differentiable (not necessarily convex and without assuming any Lipschitz continuity of the corresponding gradient mapping). The method itself is essentially known and may be viewed as a special instance of the method described in Tseng and Yun [37], see also the PhD Thesis by Milzarek [25]. This version differs from the orignal one in [16] and its variants considered for convex problems by using a different line search globalization strategy. The proximal gradient method described here plays a central role in the globalization of our proximal Newton-type method.

To motivate the proximal gradient method, let us first recall that the classical (weighted) gradient method for the minimization of a smooth objective function f first computes a minimizer d^k of the quadratic subproblem

$$\min_d f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T H_k d \tag{4}$$

for some $H_k \succ 0$, and then takes $x^{k+1} = x^k + t_k d^k$ for some suitable stepsize $t_k > 0$. Usually, H_k is chosen as a positive multiple of the identity matrix. For $H_k = I$, we get the method of steepest descent, hence d^k is given by $-\nabla f(x^k)$ in this case.

Next consider the composite optimization problem from (1). To solve this nonsmooth problem, we simply add the nonsmooth function to the argument of (4) and obtain the subproblem

$$\min_d f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T H_k d + \varphi(x^k + d). \tag{5}$$

Let $d^k = d_{H_k}(x^k)$ be a solution of this subproblem. The next iterate is then defined by $x^{k+1} := x^k + t_k d^k$ for a suitable stepsize $t_k > 0$. A simple calculation shows that the solution d^k of (5) is given by

$$d^k = \operatorname{prox}_{\varphi}^{H_k}\left(x^k - H_k^{-1}\nabla f(x^k)\right) - x^k. \tag{6}$$

We now state our proximal gradient method explicitly. The stepsize rule uses the expression

$$\Delta_k := \nabla f(x^k)^T d^k + \varphi(x^k + d^k) - \varphi(x^k) \tag{7}$$

for $k \in \mathbb{N}_0$, which is a kind of alternative of the directional derivative $\psi'(x^k, d^k)$, see Lemma 2.3. Occasionally, we write Δ instead of Δ_k, if it is computed in some variables x and d instead of x^k and d^k, respectively.

Algorithm 2.2 (Proximal Gradient Method)

(S.0) Choose $x^0 \in \operatorname{dom}\varphi$, $\beta, \sigma \in (0,1)$, and set $k := 0$.

(S.1) Choose $H_k \succ 0$ and determine d^k as the solution of

$$\min_d \nabla f(x^k)^T d + \frac{1}{2} d^T H_k d + \varphi(x^k + d).$$

(S.2) If $d^k = 0$: STOP.

(S.3) Compute $t_k = \max\{\beta^l : l = 0, 1, 2, \dots\}$ such that $\psi(x^k + t_k d^k) \leq \psi(x^k) + t_k \sigma \Delta_k$.

(S.4) Set $x^{k+1} := x^k + t_k d^k$, $k \leftarrow k + 1$, and go to (S.1).

The algorithm allows H_k to be any positive definite matrix. In general, it is chosen independently of the iteration and as a positive multiple of the identity matrix, because in that case the computation of the proximity operator is less costly, in some cases (depending on the mapping φ) even an explicit expression is known.

We now want to prove that Algorithm 2.2 is well-defined and justify the termination criterion. The analysis is mainly based on [25, 37]. Note that we assume implicitly that the algorithm does not terminate after finitely many steps.

We first give an estimate for the value of Δ, which is essentially [26, Lemma 3.5].

Lemma 2.3. *Let $x \in \operatorname{dom}\varphi$, $H \in \mathbb{S}^n_{++}$ be given, and set $d := \operatorname{prox}^H_\varphi\big(x - H^{-1}\nabla f(x)\big) - x$, cf. (6). Then the inequalities $\psi'(x; d) \leq \Delta \leq -d^T H d$ hold.*

Note that this result implies that Δ_k is always a negative number as long as d^k is nonzero. The case $d^k = 0$ is discussed in the following result, which also justifies the termination criterion in (S.2). Its proof coincides with [21, Proposition 2.5].

Lemma 2.4. *Let $H_k \in \mathbb{S}^n_{++}$ be given. Then $x^k \in \operatorname{dom}\varphi$ is a stationary point of ψ if and only if $d^k = 0$.*

Thus, the termination criterion in (S.2) of Algorithm 2.2 ensures that the algorithm terminates in a stationary point of ψ. Together with the next result, it follows that Algorithm 2.2 is well-defined.

Corollary 2.5. *Algorithm 2.2 is well-defined, and we have $\psi(x^{k+1}) < \psi(x^k)$ for all k.*

Proof. Consider a fixed iteration index k. Since, by assumption, the algorithm generates an infinite sequence, (S.2) yields $d^k \neq 0$ for all k. Thus, by Lemma 2.3, we have $\Delta_k < 0$. Using the first inequality in Lemma 2.3, we therefore obtain

$$\frac{\psi(x^k + td^k) - \psi(x^k)}{t} \leq \sigma\Delta_k$$

for all sufficiently small $t > 0$. Rearranging this inequality, we see that the step size rule (S.3) and, consequently, the whole algorithm is well-defined. Furthermore, using $\Delta_k < 0$ in (S.3) yields $\psi(x^{k+1}) = \psi(x^k + t_k d^k) \leq \psi(x^k) + t_k\sigma\Delta_k < \psi(x^k)$, and this completes the proof. □

The following convergence result is a special case of the corresponding theorem in [37].

Theorem 2.6. *Let $\{H_k\}_k \subset \mathbb{S}^n_{++}$ be a sequence such that there exist $0 < m < M$ with $mI \preceq H_k \preceq MI$ for all $k \in \mathbb{N}_0$. Then any accumulation point of a sequence generated by Algorithm 2.2 is a stationary point of ψ.*

Theorem 2.6 cannot be applied directly in order to verify global convergence of our inexact proximal Newton-type method since only some of the search directions d^k are computed by a proximal gradient method, whereas other directions correspond to an inexact proximal Newton-type step. However, a closer inspection of the proof of Theorem 2.6 yields that the following slightly stronger convergence result holds.

Remark 2.7. An easy consequence of the proof of Theorem 2.6 is the following more general result: Let $\{x^k\}$ be a sequence such that $x^{k+1} = x^k + t_k d^k$ holds for all k with some search directions $d^k \in \mathbb{R}^n$ (not necessarily generated by a proximal gradient step) and a stepsize $t_k > 0$. Assume further that $\psi(x^{k+1}) \leq \psi(x^k)$ holds for all k. Let $\{x^k\}_K$ be a convergent subsequence of the given sequence such that the search directions $d^k = d_{H_k}(x^k)$ are obtained by proximal gradient steps for all $k \in K$, where $mI \preceq H_k \preceq MI$ $(0 < m \leq M)$, and the corresponding step sizes $t_k > 0$ are determined by the Armijo-type rule from (S.3). Then the limit point of the subsequence $\{x^k\}_K$ is still a stationary point of ψ. ◊

3 Globalized Inexact Proximal Newton-type Method

Let us start with the derivation of our globalized inexact proximal Newton-type method. To this end, let us first assume that H_k stands for the exact Hessian $\nabla^2 f(x^k)$ (later H_k will be allowed to be an approximation of the Hessian only).

In smooth optimization, one step of the classical version of Newton's method for the minimizing of a function $f : \mathbb{R}^n \to \mathbb{R}$ consists in finding a solution of $H_k(x - x^k) = -\nabla f(x^k)$. This is equivalent (assuming H_k being positive definite for the moment) to solve the problem $\min_x f_k(x)$, where

$$f_k(x) := f(x^k) + \nabla f(x^k)^T(x - x^k) + \tfrac{1}{2}(x - x^k)^T H_k(x - x^k) \tag{8}$$

is a quadratic approximation of f at the current iterate x^k. To solve this problem inexactly, one often uses the criterion

$$\|\nabla f_k(x)\| \leq \eta_k \|\nabla f(x^k)\| \tag{9}$$

for some $\eta_k \in (0,1)$.

Now we adapt this strategy to the nonsmooth problem (1). In this case, the objective function is $f + \varphi$, and the corresponding approximation we use is

$$\psi_k(x) := f_k(x) + \varphi(x) = f(x^k) + \nabla f(x^k)^T(x - x^k) + \tfrac{1}{2}(x - x^k)^T H_k(x - x^k) + \varphi(x). \tag{10}$$

In view of Lemma 2.4, we may view

$$F(x) := x - \operatorname{prox}_{\varphi}\big(x - \nabla f(x)\big) \tag{11}$$

as a replacement for the derivative of the objective function since $F(x) = 0$ if and only if x is a stationary point of ψ.

Since ψ_k is another function of the form (1), one can use the same idea to replace the derivative of ψ_k by

$$F^k(x) := x - \operatorname{prox}_{\varphi}\big(x - \nabla f_k(x)\big) = x - \operatorname{prox}_{\varphi}\big(x - (\nabla f(x^k) + H_k(x - x^k))\big).$$

This observation motivates to replace the inexactness criterion (9) by a condition like $\|F^k(x)\| \leq \eta_k \|F(x^k)\|$ for some $\tau > 0$ and $\eta_k \geq 0$, see [7, 21].

The main idea of our globalized proximal Newton-type method is now similar to a standard globalization of the classical Newton method for smooth unconstrained optimization problems: Whenever the proximal Newton-type direction exists and satisfies a suitable sufficient decrease condition, the proximal Newton-type direction is accepted and followed by a line search. Otherwise, a proximal gradient step is taken which always exists and guarantees suitable global convergence properties. The descent criterion used here is motivated by the condition in [12,30]. The line search is based on the Armijo-type condition already used in the proximal gradient method and makes use of the same Δ_k that was already defined in (7). The exact statement of our method is as follows, where, now, we allow H_k to be an approximation of the Hessian of f at x^k.

Algorithm 3.1 (Globalized Inexact Proximal Newton-type Method (GIPN))

(S.0) *Choose initial parameters:* $x^0 \in \operatorname{dom}\varphi$, $\rho > 0$, $p > 2$, $,\beta, \eta \in (0,1)$, $\sigma \in (0, \frac{1}{2})$, $\zeta \in (\sigma, \frac{1}{2})$, $0 < c_{\min} \leq c_{\max}$, *and set* $k := 0$.

(S.1) *Choose* $H_k \in \mathbb{R}^{n\times n}$ *symmetric,* $\eta_k \in [0, \eta)$ *and compute an inexact solution* $\hat{x}^k$ *of the subproblem* $\min_x \psi_k(x)$ *satisfying*

$$\|F^k(\hat{x}^k)\| \leq \eta_k \|F(x^k)\| \qquad \text{and} \qquad \psi_k(\hat{x}^k) - \psi_k(x^k) \leq \zeta \Delta_k, \tag{12}$$

and set $d^k := \hat{x}^k - x^k$. *If this is not possible or the condition*

$$\Delta_k \leq -\rho \|d^k\|^p \tag{13}$$

is not satisfied, choose $c_k \in [c_{\min}, c_{\max}]$ *and determine* d^k *as the (unique) solution of*

$$\min_d \nabla f(x^k)^T d + \frac{1}{2} c_k \|d\|^2 + \varphi(x^k + d).$$

(S.2) If $d^k = 0$: STOP.

(S.3) Compute $t_k = \max\{\beta^l \mid l = 0, 1, 2, \dots\}$ such that $\psi(x^k + t_k d^k) \leq \psi(x^k) + \sigma t_k \Delta_k$.

(S.4) Set $x^{k+1} := x^k + t_k d^k$, $k \leftarrow k + 1$ and go to (S.1).

Before we start to analyse the convergence properties of Algorithm 3.1, let us add a few comments regarding the proximal subproblems that we try to solve inexactly in (S.1). Since H_k is not necessarily positive definite, these subproblems are not guaranteed to have a solution. The same difficulty arises within the classical Newton method since, in the indefinite case, the quadratic subproblem (8) certainly has no minimizer. Nevertheless, the classical Newton method is often quite successful even if H_k is indefinite (at least during some intermediate iterations), and the Newton direction is usually well-defined because it just computes a stationary point of the subproblem (8) which exists also for indefinite matrices H_k. Here, the situation is similar since the conditions (12) only check whether we have an (inexact) stationary point (note that these conditions certainly hold for the exact solution of the corresponding subproblem, cf. [21, Proposition 2.4] for the second condition and note that $\zeta < \frac{1}{2}$). Moreover, the situation here is even better than in the classical case since the additional function φ may guarantee the existence of a minimum even for indefinite H_k (e.g. if φ has compact support as this occurs when φ is the indicator function of a bounded feasible set). We therefore believe that our proximal Newton-type direction does exist in many situations (otherwise we switch to the proximal gradient direction).

The properties of Algorithm 3.1 obviously depend on the choice of the matrices H_k and the degree of inexactness that is used to compute the inexact proximal Newton-type direction in (S.1). This degree is specified by the test in (12). The local convergence analysis requires some additional conditions regarding the choice of the sequence η_k, whereas the global convergence analysis depends only on the choice $\eta_k \in [0, \eta)$ for some given $\eta \in (0, 1)$ and does not need the second condition in (12). The condition in (13) is a sufficient decrease condition, with $\rho > 0$ typically being a small constant.

For our subsequent analysis, we set

$$\mathcal{K}_G := \{k : x^{k+1} \text{ was generated by the proximal gradient method}\},$$
$$\mathcal{K}_N := \{k : x^{k+1} \text{ was generated by the inexact proximal Newton-type method}\}.$$

The following result shows that the step size rule in (S.3) is well-defined and Algorithm 3.1 is a descent method.

Proposition 3.2. *Consider a fixed iteration k and suppose that $d^k \neq 0$. Then the line search in (S.3) is well-defined and yields a new iterate x^{k+1} satisfying $\psi(x^{k+1}) < \psi(x^k)$.*

Proof. Since the proximal gradient method is well-defined by Corollary 2.5, the claim holds for $k \in \mathcal{K}_G$. Now, assume $k \in \mathcal{K}_N$, in which case (13) holds. Then $\Delta_k < 0$ and, therefore, the remaining part of the proof is identical to the one of Corollary 2.5. □

Proposition 3.2 requires $d^k \neq 0$. In view of the following result, this assumption can be stated without loss of generality. In particular, this result justifies our termination criterion in (S.2).

Lemma 3.3. *An iterate x^k generated by* GIPN *is a stationary point of ψ if and only if $d^k = 0$.*

Proof. For $k \in \mathcal{K}_G$, the result follows from Lemma 2.4. Hence assume $k \in \mathcal{K}_N$, and let $d^k = 0$. This yields $\hat{x}^k = x^k$. Since $F^k(x^k) = F(x^k)$, condition (12) yields $\|F(x^k)\| \leq \eta_k \cdot \|F(x^k)\|$. As $\eta_k \in [0, 1)$, we get $F(x^k) = 0$ and x^k is a stationary point of ψ, using again Lemma 2.4. Conversely, assume that $d^k \neq 0$ for $k \in \mathcal{K}_N$. Then, analogous to Lemma 2.3, we get $\psi'(x^k; d^k) \leq \Delta_k \leq -\rho\|d^k\|^p < 0$. Hence x^k is not a stationary point of ψ. □

Altogether, the previous results show that Algorithm 3.1 is well-defined.

4 Convergence Theory

In the following, we will prove global and local convergence results for algorithm GIPN. For this purpose, we assume that GIPN generates an infinite sequence and $d^k \neq 0$ holds for all $k \in \mathbb{N}$. The latter is motivated by Lemma 3.3.

4.1 Global Convergence

The following is the main global convergence result for Algorithm 3.1. It guarantees stationarity of any accumulation point. Hence, if f is also convex, this implies that any accumulation point is a solution of the composite optimization problem from (1).

Theorem 4.1. *Consider Algorithm GIPN with a bounded sequence of matrices $\{H_k\}$. Then every accumulation point of a sequence generated by this method is a stationary point of ψ.*

Proof. Let $\{x^k\}$ be a sequence generated by GIPN and $\{x^k\}_K$ a subsequence of $\{x^k\}$ converging to some x^*. If there are infinitely many indices $k \in K$ with $k \in \mathcal{K}_G$, i.e. the subsequence contains infinitely many iterates x^k such that x^{k+1} is generated by the proximal gradient method, Proposition 3.2 and the statement of Remark 2.7 yield that x^* is a stationary point of ψ.

Hence consider the case where all elements of the subsequence $\{x^{k+1}\}_K$ are generated by inexact Newton-type steps. Since $\{\psi(x^k)\}$ is monotonically decreasing by Proposition 3.2, $\{x^k\}_K$ converges to x^*, and since ψ is lsc, we get the convergence of the entire sequence $\{\psi(x^k)\}$ to some finite number ψ^*. The line search rule therefore yields

$$0 \leftarrow \psi(x^{k+1}) - \psi(x^k) \leq \sigma t_k \Delta_k < 0$$

and, hence, $t_k\Delta_k \to 0$ for $k \to \infty$. We claim that this implies $\{\|d^k\|\}_K \to 0$ (possibly after taking another subsequence). To verify this statement, we distinguish two cases:

Case 1: $\liminf_{k\in K} t_k > 0$. Then $\{\Delta_k\}_K \to 0$, and we therefore obtain $\{\|d^k\|\}_K \to 0$ in view of (13).

Case 2: $\liminf_{k\in K} t_k = 0$. Without loss of generality, assume $\lim_{k\in K} t_k = 0$. Then, for all $k \in K$ sufficiently large, the line search test is violated for the stepsize $\tau_k := t_k/\beta$. Using the monotonicity of the difference quotient of convex functions, cf. [3, Proposition 9.27], and the definition of Δ_k, we therefore obtain

$$\begin{aligned}\sigma\Delta_k &< \frac{\psi(x^k + \tau_k d^k) - \psi(x^k)}{\tau_k} \leq \frac{f(x^k + \tau_k d^k) - f(x^k)}{\tau_k} + \varphi(x^k + d^k) - \varphi(x^k) \\ &= \frac{f(x^k + \tau_k d^k) - f(x^k)}{\tau_k} - \nabla f(x^k)^T d^k + \Delta_k = \big(\nabla f(\xi^k) - \nabla f(x^k)\big)^T d^k + \Delta_k\end{aligned}$$

for all $k \in K$ sufficiently large, where the last expression uses the mean value theorem with some $\xi^k \in (x^k, x^k + \tau_k d^k)$. Reordering these expressions, we obtain

$$0 < -(1-\sigma)\Delta_k < \big(\nabla f(\xi^k) - \nabla f(x^k)\big)^T d^k.$$

Using (13) we get $(1-\sigma)\rho\|d^k\|^{p-1} \leq \|\nabla f(\xi^k) - \nabla f(x^k)\|$ for all $k \in K$. Since $x^k \to_K x^*$, $\tau_k \to_K 0$, and $\{d^k\}_K$ is necessarily bounded (as a consequence of (13) with $p > 1$), it follows that the right-hand side converges to zero. This implies $d^k \to_K 0$.

Therefore, $d^k \to_K 0$ holds in both cases. Since $x^k \to_K x^*$, the definition of d^k also implies $\hat{x}^k \to_K x^*$. Using the continuity of the proximity operator, we therefore get

$$F(x^k) \to_K x^* - \operatorname{prox}_\varphi\big(x^* - \nabla f(x^*)\big)$$

and, since $\{H_k\}$ is bounded by assumption,

$$F^k(\hat{x}^k) \to_K x^* - \operatorname{prox}_\varphi\big(x^* - \nabla f(x^*)\big).$$

Since $\|F^k(\hat{x}^k)\| \leq \eta\|F(x^k)\|$ for all $k \in K$ in view of (12) and $\eta \in (0,1)$, taking the limit $k \to_K \infty$ therefore implies $x^* = \operatorname{prox}_\varphi\big(x^* - \nabla f(x^*)\big)$, which is equivalent to x^* being a stationary point of ψ. □

Remark 4.2. The proof of Theorem 4.1 also verifies $\{d^k\}_{\mathcal{K}_N} \to 0$ if the sequence $\{\psi(x^k)\}$ is bounded below and therefore convergent. In particular, this is satisfied whenever the sequence $\{x^k\}$ has an accumulation point. ◊

Remark 4.3. Note that the proof of Theorem 4.1 only requires $p > 1$ and the first condition from (12). The stronger or additional conditions are only needed in the local convergence theory. ◊

4.2 Local Convergence

We now turn to the local convergence properties of Algorithm 3.1. To this end, we assume that f is twice continuously differentiable and the sequence $\{H_k\}$ is bounded and satisfies the Dennis-Moré condition [13]

$$\lim_{k\to\infty} \frac{\big\|\big(H_k - \nabla^2 f(x^*)\big)(\hat{x}^k - x^k)\big\|}{\|\hat{x}^k - x^k\|} = 0.$$

Under suitable assumptions, we expect the method to be locally superlinearly or quadratically convergent. The main steps into this direction are summarized in the following observations, which are partly taken from [39].

Proposition 4.4. *Consider Algorithm 3.1 with $\{H_k\}$ satisfying the Dennis-Moré condition and $MI \succeq H_k \succeq mI$ for all $k \in \mathbb{N}_0$ with suitable $M \geq m > 0$. Let x^* be a stationary point of ψ such that $\nabla^2 f(x^*)$ is positive definite. Then there exist constants $\varepsilon > 0$ as well as $C, \kappa_1, \kappa_2, \mu > 0$ such that, for any iterate $x^k \in B_\varepsilon(x^*)$, the following statements hold, where $\hat{x}^k_{ex}$ is the exact solution of the corresponding subproblem in (S.1) of Algorithm 3.1:*

(a) $\big\|\hat{x}^k - \hat{x}^k_{ex}\big\| \leq C\eta_k \|F(x^k)\|$.

(b) $\big\|\hat{x}^k_{ex} - x^k\big\| \leq \kappa_1 \|x^k - x^*\|$.

(c) $\big\|x^k - x^*\| \leq \kappa_2 \big\|F(x^k)\big\|$.

(d) $\big\|\hat{x}^k_{ex} - x^*\big\| \leq \frac{1}{\mu}\Big(\big\|\nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^*)(x^k - x^*)\big\| + \big\|\big(H_k - \nabla^2 f(x^*)\big)(\hat{x}^k_{ex} - x^k)\big\|\Big)$.

(e) The search direction $d^k = \hat{x}^k - x^k$ from (S.1) satisfies the sufficient decrease condition (13).

(f) The full stepsize $t_k = 1$ satisfies the Armijo-type condition from (S.3).

Proof. First note that the assumed positive definiteness of $\nabla^2 f(x^*)$ implies, possibly by changing the values of m, M, that

$$m\|d\|^2 \leq d^T \nabla^2 f(x) d \leq M\|d\|^2 \quad \forall d \in \mathbb{R}^n, \ \forall x \in B_\varepsilon(x^*). \tag{14}$$

Throughout this proof, we assume that the given iterate x^k belongs to this neighbourhood $B_\varepsilon(x^*)$. We now verify each of the six statements separately (using a possibly smaller radius ε).

(a) First, note that the function ψ_k is strongly convex and, therefore, has a unique minimizer. Thus, the exact solution of the subproblem exists and hence guarantees that there is an inexact solution $\hat{x}^k$.

Since $F^k(\hat{x}^k) = \hat{x}^k - \operatorname{prox}_\varphi\big(\hat{x}^k - \nabla f_k(\hat{x}^k)\big)$, we obtain from (3) that

$$F^k(\hat{x}^k) - \nabla f_k(\hat{x}^k) \in \partial\varphi(\hat{x}^k - F^k(\hat{x}^k)).$$

The definition of ψ_k together with the subdifferential sum rule therefore implies

$$F^k(\hat{x}^k) + \nabla f_k\big(\hat{x}^k - F^k(\hat{x}^k)\big) - \nabla f_k(\hat{x}^k) \in \partial\psi_k\big(\hat{x}^k - F^k(\hat{x}^k)\big)$$

which is equivalent to

$$(I - H_k)F^k(\hat{x}^k) \in \partial\psi_k\big(\hat{x}^k - F^k(\hat{x}^k)\big). \tag{15}$$

Since ψ_k is strongly convex in $B_\varepsilon(x^*)$ with modulus $m > 0$, its subdifferential is strongly monotone in this neighbourhood with the same modulus. Hence, using (15) together with $0 \in \partial\psi_k(\hat{x}^k_{ex})$, we get

$$\big\langle (I - H_k)F^k(\hat{x}^k), \hat{x}^k - F^k(\hat{x}^k) - \hat{x}^k_{ex} \big\rangle \geq m \big\| \hat{x}^k - F^k(\hat{x}^k) - \hat{x}^k_{ex} \big\|^2.$$

Applying the Cauchy-Schwarz inequality, this implies

$$\big\| \hat{x}^k - F^k(\hat{x}^k) - \hat{x}^k_{ex} \big\| \leq \frac{1}{m} \big\| (I - H_k)F^k(\hat{x}^k) \big\| \leq \frac{1}{m}(1 + M)\|F^k(\hat{x}^k)\|.$$

Using the inexactness criterion (12), we finally get, with $C := (1 + M + m)/m$,

$$\begin{aligned} \|\hat{x}^k - \hat{x}^k_{ex}\| &\leq \|\hat{x}^k - F^k(\hat{x}^k) - \hat{x}^k_{ex}\| + \|F^k(\hat{x}^k)\| \\ &\leq \frac{1}{m}(1 + M)\|F^k(\hat{x}^k)\| + \|F^k(\hat{x}^k)\| \leq C\eta_k \|F(x^k)\|. \end{aligned}$$

(b) Let $G(x, H) := x - \operatorname{prox}^H_\varphi \big(x - H^{-1}\nabla f(x)\big)$. By Lemma 2.1, G is Lipschitz continuous for $x \in B_\varepsilon(x^*)$ and $H \in \mathbb{S}^n_{++}$ with $mI \preceq H \preceq MI$ and $G(x^*, H) = 0$ for all such H by Lemma 2.4. Thus, there exists $\kappa_1 > 0$ (not depending on H_k) such that

$$\|\hat{x}^k_{ex} - x^k\| = \|G(x^k, H_k)\| = \|G(x^k, H_k) - G(x^*, H_k)\| \leq \kappa_1 \|x^k - x^*\|.$$

(c) Reducing $\varepsilon > 0$ if necessary, the twice continuous differentiability of f implies that we can choose a convex neighbourhood $B_\varepsilon(x^*)$ of x^* such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \tag{16}$$

for all $x, y \in B_\varepsilon(x^*)$ with some Lipschitz constant $L > 0$. Since (14) implies the strong convexity of f in $B_\varepsilon(x^*)$, it follows that ∇f is strongly monotone in $B_\varepsilon(x^*)$ with

$$m\|x - y\|^2 \leq \big\langle \nabla f(x) - \nabla f(y), x - y \big\rangle \tag{17}$$

for all $x, y \in B_\varepsilon(x^*)$. Now, let $x^k \in B_\varepsilon(x^*)$ be arbitrary and set $u^k := F(x^k)$, i.e. $x^k - u^k = \operatorname{prox}_\varphi \big(x^k - \nabla f(x^k)\big)$. Since x^* is a stationary point of ψ, we have $x^* = \operatorname{prox}_\varphi \big(x^* - \nabla f(x^*)\big)$. Recall that the proximity operator is firmly nonexpansive, meaning that

$$\|\operatorname{prox}_\varphi(a) - \operatorname{prox}_\varphi(b)\|^2 \leq \big\langle \operatorname{prox}_\varphi(a) - \operatorname{prox}_\varphi(b), a - b \big\rangle$$

for all $a, b \in \operatorname{dom}(\varphi)$. Setting $a := x^k - \nabla f(x^k), b := x^* - \nabla f(x^*)$, we therefore obtain

$$\|x^k - u^k - x^*\|^2 \leq \big\langle x^k - u^k - x^*, x^k - \nabla f(x^k) - x^* + \nabla f(x^*) \big\rangle.$$

After some algebraic manipulations and cancellation, this can be rewritten as

$$\big\langle x^k - x^*, \nabla f(x^k) - \nabla f(x^*) \big\rangle \leq \big\langle u^k, (x^k - x^*) + (\nabla f(x^k) - \nabla f(x^*) \big\rangle - \|u^k\|^2.$$

Applying (16) and (17) together with the Cauchy-Schwarz inequality, we obtain

$$m\|x^k - x^*\|^2 \leq \|u^k\| \big(\|x^k - x^*\| + \|\nabla f(x^k) - \nabla f(x^*)\| \big) \leq (1 + L)\|u^k\| \, \|x^k - x^*\|.$$

Therefore, $\|x^k - x^*\| \leq \frac{1+L}{m}\|F(x^k)\|$.

(d) The inequality holds trivially for $\hat{x}^k_{ex} = x^*$. Thus, assume $\hat{x}^k_{ex} \neq x^*$. Using stationarity of x^* and $\hat{x}^k_{ex}$, we have $0 \in \nabla f(x^*) + \partial\varphi(x^*)$ and $0 \in \nabla f(x^k) + H_k(\hat{x}^k_{ex} - x^k) + \partial\varphi(\hat{x}^k_{ex})$. By the monotonicity of the subdifferential of φ, we get

$$\begin{aligned}
0 \leq & \big\langle \nabla f(x^k) - \nabla f(x^*) + H_k(\hat{x}^k_{ex} - x^k), x^* - \hat{x}^k_{ex} \big\rangle \\
= & \big\langle \nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^*)(x^k - x^*), x^* - \hat{x}^k_{ex} \big\rangle + \big\langle \nabla^2 f(x^*)(\hat{x}^k_{ex} - x^*), x^* - \hat{x}^k_{ex} \big\rangle \\
& + \big\langle \big(\nabla^2 f(x^*) - H_k\big)(x^k - \hat{x}^k_{ex}), x^* - \hat{x}^k_{ex} \big\rangle \\
\leq & \big\|\nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^*)(x^k - x^*)\big\| \cdot \big\|x^* - \hat{x}^k_{ex}\big\| - m\big\|\hat{x}^k_{ex} - x^*\big\|^2 \\
& + \big\|\big(\nabla^2 f(x^*) - H_k\big)(x^k - \hat{x}^k_{ex})\big\| \cdot \big\|x^* - \hat{x}^k_{ex}\big\|.
\end{aligned}$$

Rearranging terms yields

$$\big\|\hat{x}^k_{ex} - x^*\big\| \leq \frac{1}{\mu}\Big(\|\nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^*)(x^k - x^*)\| + \big\|\big(\nabla^2 f(x^*) - H_k\big)(x^k - \hat{x}^k_{ex})\big\|\Big)$$

with $\mu = m$.

(e) Let $\Delta_{k,N}$ be the Δ-function corresponding to the search direction $d^k_N := \hat{x}^k - x^k$, i.e. $\Delta_{k,N} := \nabla f(x^k)^T d^k_N + \varphi(x^k + d^k_N) - \varphi(x^k)$. Then the second condition in (12) is equivalent to

$$(1 - \zeta)\Delta_{k,N} \leq -\frac{1}{2}(d^k_N)^T H_k d^k_N,$$

which yields

$$\Delta_{k,N} \leq -\tilde{c}\|d^k_N\|^2 \quad \text{for} \quad \tilde{c} := m/(2(1 - \zeta)). \tag{18}$$

Since x^* is a stationary point of ψ, hence $F(x^*) = 0$, it follows from the continuity of F and the results in parts (a) and (b) that we can reduce the neighbourhood $\varepsilon > 0$ further to obtain

$$\|\hat{x}^k - \hat{x}^k_{ex}\| \leq \frac{1}{2}\left(\frac{\rho}{\tilde{c}}\right)^{1/(2-p)}, \quad \|\hat{x}^k_{ex} - x^k\| \leq \frac{1}{2}\left(\frac{\rho}{\tilde{c}}\right)^{1/(2-p)}.$$

Combining these inequalities yields $\|d^k_N\| = \|\hat{x}^k - x^k\| \leq (\rho/\tilde{c})^{1/(2-p)}$. We therefore get

$$\Delta_{k,N} \leq -\tilde{c}\|d^k_N\|^2 = -\tilde{c}\|d^k_N\|^p\|d^k_N\|^{2-p} \leq -\rho\|d^k_N\|^p.$$

Therefore, the sufficient descent condition (13) is fulfilled and the search direction $d^k = d^k_N$ is obtained by the inexact proximal Newton-type method.

(f) Taylor expansion yields

$$\begin{aligned}
f(\hat{x}^k) - f(x^k) \;=\; & \nabla f(x^k)^T(\hat{x}^k - x^k) + \frac{1}{2}(\hat{x}^k - x^k)^T\nabla^2 f(x^k)(\hat{x}^k - x^k) \\
& + \frac{1}{2}(\hat{x}^k - x^k)^T\big(\nabla^2 f(\xi^k) - \nabla^2 f(x^k)\big)(\hat{x}^k - x^k)
\end{aligned}$$

for some $\xi^k \in (x^k, \hat{x}^k)$. Hence, we get

$$\begin{aligned}
&\psi(\hat{x}^k) - \psi(x^k) + \psi_k(x^k) - \psi_k(\hat{x}^k) \\
&\quad = f(\hat{x}^k) - f(x^k) - \nabla f(x^k)^T(\hat{x}^k - x^k) - \frac{1}{2}(\hat{x}^k - x^k)^T H_k(\hat{x}^k - x^k) \\
&\quad \leq \frac{1}{2}\big\|\nabla^2 f(\xi^k) - \nabla^2 f(x^k)\big\| \cdot \|\hat{x}^k - x^k\|^2 + \frac{1}{2}\big\|\nabla^2 f(x^k) - \nabla^2 f(x^*)\big\| \cdot \|\hat{x}^k - x^k\|^2 \\
&\qquad + \frac{1}{2}\big\|\big(H_k - \nabla^2 f(x^*)\big)(\hat{x}^k - x^k)\big\| \cdot \|\hat{x}^k - x^k\|.
\end{aligned}$$

By the Dennis-Moré criterion this is $o(\|\hat{x}^k - x^k\|^2)$ for $x^k \to x^*$. As before, it follows from the continuity of F and the results in parts (a) and (b) that $\|\hat{x}^k - x^k\| \to 0$. Thus, we can reduce the neighbourhood $\varepsilon > 0$ further to obtain (using (12))

$$\begin{aligned}
\psi(\hat{x}^k) - \psi(x^k) &= \big(\psi(\hat{x}^k) - \psi(x^k) + \psi_k(x^k) - \psi_k(\hat{x}^k)\big) + \psi_k(\hat{x}^k) - \psi_k(x^k) \\
&\leq (\zeta - \sigma)\tilde{c}\|\hat{x}^k - x^k\|^2 + \zeta\Delta_k \\
&= (\zeta - \sigma)\tilde{c}\|\hat{x}^k - x^k\|^2 + \sigma\Delta_k + (\zeta - \sigma)\Delta_k \\
&\leq (\zeta - \sigma)\tilde{c}\|\hat{x}^k - x^k\|^2 + \sigma\Delta_k - (\zeta - \sigma)\tilde{c}\|\hat{x}^k - x^k\|^2 = \sigma\Delta_k,
\end{aligned}$$

where the last inequality follows from (18) (note that $\Delta_k = \Delta_{k,N}$ in the current situation). This proves that in this case the full step length is attained. □

A suitable combination of the previous results leads to the following (global and) local convergence result for Algorithm 3.1.

Theorem 4.5. *Consider Algorithm 3.1 and assume that the sequence $\{H_k\}$ satisfies the assumptions from Proposition 4.4. Let x^* be an accumulation point of a sequence $\{x^k\}$ generated by Algorithm 3.1 such that $\nabla^2 f(x^*)$ is positive definite. Then the following statements hold:*

(a) The whole sequence $\{x^k\}$ converges to x^*, and x^* is a strict local minimum of ψ.

(b) For all sufficiently large k, the search direction is attained by the inexact proximal Newton-type direction.

(c) For all sufficiently large k, the full step size $t_k = 1$ is accepted.

(d) If $\eta \leq \overline{\eta}$ for some $\overline{\eta} > 0$, the sequence $\{x^k\}$ converges linearly to x^*.

(e) If $\{\eta_k\} \to 0$, the sequence $\{x^k\}$ converges superlinearly to x^*.

Proof. In view of Theorem 4.1, every accumulation point of the sequence $\{x^k\}$ is a stationary point of ψ. Hence we can apply Proposition 4.4. Furthermore, the assumed positive definiteness of $\nabla^2 f(x^*)$ implies that ψ is locally strongly convex and, therefore, has x^* as the only stationary point in a suitable neighbourhood. Hence x^* is necessarily the only accumulation point of the sequence $\{x^k\}$ in this neighbourhood. In order to verify statement (a), we therefore have to verify only the condition $\{\|x^{k+1} - x^k\|\}_K \to 0$ for any subsequence $\{x^k\}_K \to x^*$, cf. [27, Lemma 4.10].

Hence let $\{x^k\}_K$ denote an arbitrary subsequence converging to x^*. Since $\|x^{k+1}-x^k\| = t_k\|d^k\| \leq \|d^k\|$ for all $k \in \mathbb{N}$, it suffices to show $\{\|d^k\|\}_K \to 0$ for $K \subset \mathcal{K}_G$ and $K \subset \mathcal{K}_N$. If $K \subset \mathcal{K}_G$, this follows from the continuity of the solution operator in the proximal gradient method, see Lemma 2.1. On the other hand, if $K \subset \mathcal{K}_N$, the statement is already noted in Remark 4.2. Thus, we proved part (a). Statements (b) and (c) follow from Proposition 4.4.

For the remaining part choose $\varepsilon > 0$ such that Proposition 4.4 holds for $x^k \in B_\varepsilon(x^*)$ and ∇f is Lipschitz continuous with constant $L > 0$ in $B_\varepsilon(x^*)$. Let k_0 be sufficiently large such that all iterates x^k for $k \geq k_0$ are in this neighbourhood. Note that

$$\begin{aligned}\|F(x^k)\| &= \|x^k - \operatorname{prox}_\varphi(x^k - \nabla f(x^k))\| \\ &= \|x^k - \operatorname{prox}_\varphi(x^k - \nabla f(x^k)) - x^* + \operatorname{prox}_\varphi(x^* - \nabla f(x^*))\| \\ &\leq 2\|x^k - x^*\| + \|\nabla f(x^k) - \nabla f(x^*)\| \leq (2+L)\|x^k - x^*\|,\end{aligned}$$

where the inequality uses the nonexpansivity of the proximity operator, cf. [11, Lemma 2.4] By Proposition 4.4 (a) and (d), we get

$$\begin{aligned}\|x^{k+1} - x^*\| = \|\hat{x}^k - x^*\| \;&\leq\; \|\hat{x}^k - \hat{x}^k_{ex}\| + \|\hat{x}^k_{ex} - x^*\| \\ &\leq C\eta_k\|F(x^k)\| + \frac{1}{\mu}\|\nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^*)(x^* - x^k)\| \\ &\quad + \frac{1}{\mu}\big\|\big(H_k - \nabla^2 f(x^*)\big)(\hat{x}^k_{ex} - x^k)\big\|.\end{aligned}$$

The twice continuous differentiability of f yields that the second term is $o(\|x^k - x^*\|)$. The Dennis-Moré condition implies that the third term is $o(\|x^k - x^*\|)$. Thus, the above yields parts (d) for $\overline{\eta} < 1/(C(L+2))$. Finally, under the assumptions of part (e), also the first term is $o(\|x^k - x^*\|)$, which completes the proof. □

Note that one can also verify local quadratic convergence under slightly stronger assumption as in Theorem 4.5 (e), in particular, using a stronger version of the Dennis-Moré condition. The details are left to the reader.

5 Numerical Results

In this section we report some numerical results for solving problem (1) and show the competitiveness compared to several state-of-the-art methods. All methods are implemented in Matlab.

In the following, GPN denotes the globalized (inexact) proximal Newton method, whereas QGPN denotes a globalized (inexact) proximal quasi-Newton method, where the exact Hessian is replaced by a limited memory BFGS-update.

5.1 Logistic Regression with ℓ_1-Penalty

In this example, we consider the logistic regression problem

$$\min_{y,v} \frac{1}{m} \sum_{i=1}^{m} \log\Big(1 + \exp\big(-b_i(a_i^T y + v)\big)\Big) + \lambda \|y\|_1, \tag{19}$$

where $a_i \in \mathbb{R}^n$ $(i = 1, \dots, m)$ are given feature vectors and $b_i \in \{\pm 1\}$ the corresponding labels, $\lambda > 0$, $y \in \mathbb{R}^n, v \in \mathbb{R}$. Usually, we have $m \gg n$. Logistic regression is used to separate data by a hyperplane, see [19] for further information.

With $\phi\colon \mathbb{R} \to \mathbb{R}$, $\phi(u) := \log\big(1 + \exp(-u)\big)$, $x := (y^T, v)^T$ and $A \in \mathbb{R}^{m \times (n+1)}$, where the i-th row of A is $(b_i a_i^T, b_i)$ for $i = 1, \dots, m$, we can write (19) equivalently as

$$\min_x \psi(x) := \frac{1}{m} \sum_{i=1}^{m} \phi\big((Ax)_i\big) + \lambda \|x_{\{1,\dots,n\}}\|_1. \tag{20}$$

The function ϕ is convex, but not strictly convex, and its derivative is globally Lipschitz continuous. Thus, this holds also for the smooth part of ψ.

5.1.1 Algorithmic Details

Subproblem solvers. The crucial part of the implementation is the solution of the subproblem (12). We use two methods for this aim, which are described below: The fast iterative shrinkage thresholding algorithm (FISTA) [5] and the globalized semismooth Newton method (SNF) [26].

Since the inexact termination criterion (12) is not practicable without significant additional computation costs, we minimize (10) using the standard termination criterion for these solvers with a low maximal number of iterations. The tolerance is adapted in each step such that the subproblems are solved more exactly when the current iterate is near the solution.

FISTA by Beck and Teboulle [5] is an accelerated first order method for the solution of problems of type (1), where f is convex and has a Lipschitz continuous gradient. In every step a problem of type (5) is solved for $H_k = L_k I$, where L_k is an approximation to the Lipschitz constant of ∇f, which is updated by backtracking. After that, a step size is computed and the next iterate is a convex combination of the old iterate and the computed solution. For the approximation of the Lipschitz constant of f_k, we start with $L_0 := 1$ and use the increasing factor $\eta := 2$. For every subproblem, we perform at most 80 iterations. The globalized proximal Newton-type method with this subproblem solver is denoted by GPN-F.

SNF by Milzarek and Ulbrich [26] is a semismooth Newton method with filter globalization. Since the subproblems in this example are convex, we use the convex variant of the method. The semismooth Newton method is essentially applied to the equation $F(x) = 0$ with $F(x)$ defined in (11). After computing a search direction, a filter decides if the update is applied or a proximal gradient step is performed. All constants

are chosen as in [26] and we run the method with at most 10 iterations. We denote the globalized proximal Newton method with SNF subproblem solver by GPN-S.

Choice of parameters. We use the parameters $p = 2.1$ and $\rho = 10^{-8}$ for the acceptance criterion (13). The line search is performed with $\beta = 0.1$ and $\sigma = 10^{-4}$. The constant c_k for the proximal gradient step is initialized with $c_0 = 1/6$, and in each step adapted to reach the Lipschitz constant of the gradient of f.

Variant with Quasi-Newton-Update. In addition to the globalized proximal Newton method, we implemented a variant of the algorithm, where the exact Hessian in the quadratic approximation (10) is replaced by a limited memory BFGS-update with a memory of 10. The implementation follows [8]. Like before, we denote these methods by QGPN-F and QGPN-S, respectively.

5.1.2 State-of-the-art Methods

We check the above described variants of GPN against each other, but also compare them with several state-of-the-art methods, which are listed below.

PG. The proximal gradient method is described in Algorithm 2.2. It is a first order method to solve problem (1). We set $\beta = 0.1$, $\sigma = 10^{-4}$ and $H_k = c_k I$, where c_k is updated as before.

FISTA [5]. The fast iterative shrinkage thresholding algorithm is an accelerated variant of the proximal gradient method. Details were already given in Section 5.1.1.

SpaRSA [38]. SpaRSA (Sparse reconstruction by separable approximation) is another accelerated first order method to solve problem (1). The main difference to FISTA is the update of the factor c_k, which is done by a Barzilai-Borwein approach.

SNF [26]. The semismooth Newton method with filter globalization is described in 5.1.1. Similar to the subproblem solver, we apply the convex version of the method.

5.1.3 Numerical Comparison

We follow the example in [6] and generate test problems with $n = 10^4$ features and $m = 10^6$ training sets. Each feature vector a_i has approximately 10 nonzero entries, which are generated independently from a standard normal distribution. We choose $y^{\text{true}} \in \mathbb{R}^n$ with 100 nonzero entries and $v^{\text{true}} \in \mathbb{R}$, which are independently generated from standard normal distribution and define the labels as $b_i = \operatorname{sign}\left(a_i^T y^{\text{true}} + v^{\text{true}} + v_i\right)$, where the v_i $(i = 1, \ldots, m)$ are also chosen independently from a normal distribution with variance 0.1. The regularization parameter λ is set to $0.1\lambda_{\max}$, where $\lambda_{\max}$ is the smallest value such that $y^* = 0$ is a solution of (19). The derivation of this value can be found in [19]. For all methods, we start with the initial value $x^0 = 0$.

Due to the differences of the methods, the standard termination criteria of them are not a suitable choice to compare the performance. Thus, we compute the approximate

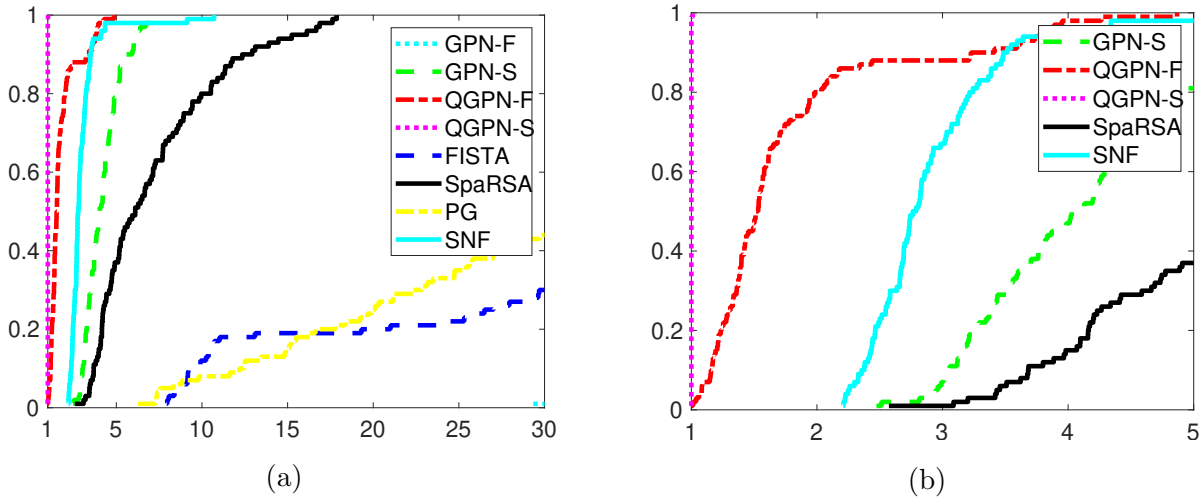


Figure 1: Performance profiles showing the runtime for 100 random test examples as described in Section 5.1.3. Figure (a) shows a range from 1 to 30 times the best method, whereas Figure (b) is scaled from 1 to 5 times the best method.

minimizer ψ^* of (19) using GPN-F with very high accuracy. We terminate each of the algorithms above when the value $\psi(x^k)$ in the current iterate x^k satisfies

$$\frac{\psi(x^k) - \psi^*}{|\psi^*|} \leq \texttt{tol} \tag{21}$$

for $\texttt{tol} = 10^{-6}$. To accomplish comparability, we look at the runtime of 100 test examples and document the results using the performance profiles introduced by Dolan and Moré [14]. The results are shown in Figure 1, the averaged values for some counters are given in Table 1.

Comparison of GPN-variants. We start with a comparison of the variants of the globalized proximal Newton-type methods, namely GPN-F, GPN-S, QGPN-F, and QGPN-S. We see that the semismooth Newton subproblem solver performs much better than the FISTA solver. One reason for this is that we can terminate the subproblem solvers in (Q)GPN-S after only 10 iterations to get reasonable results, whereas test runs show that (Q)GPN-F performs best with a maximum of 80 iterations in each subproblem. Nevertheless, note that every iteration of SNF itself needs to solve a linear system by the CG method, but both, FISTA and SNF, need to evaluate the product $\nabla^2 f(x^k)z$ for some $z \in \mathbb{R}^n$ in every iteration, which is the most expensive part of the algorithm since it involves two multiplications with A or A^T.

Furthermore, the performance of the variants with limited memory BFGS-update for the Hessian of the smooth part is significantly better than the use of the exact Hessian, although we need more outer and inner iterations to reach the termination accuracy. Again, this is due to the number of Hessian-vector-multiplications, which appear in QGPN only once in every iteration to compute the function value and the BFGS-update, whereas in GPN they are needed in every inner iteration.

Both arguments together verify why QGPN-S is the best variant tested, whereas the performance of GPN-F is not competitive.

We see in Table 1 that almost all solutions of the subproblems satisfy the descent condition (13) and, since the number of function evaluations is approximately the number of outer iterations, almost all search directions are applied with full step length. Thus, for this example, the globalization is not necessary, neither in theory nor in the numerical examples.

Comparison to other methods. Since FISTA and the proximal gradient method are first order methods, it is not surprising that they need considerable more iterations to reach the termination tolerance. Thus, with the same arguments as above, they are not competitive due to the huge number of matrix-vector-products involving the matrices A or A^T, although they do not need to evaluate the Hessians. The third first order method, SpaRSA, is far better, because the number of iterations and therefore the number of matrix-vector-products is much smaller, but it is still not able to compete with the second order methods.

The semismooth Newton method with filter globalization is the only second order method we compare our method to. As before, we see a correlation between the runtime and the number of matrix-vector-products with one of the matrices A or A^T. As this number is higher than the one of QGPN, the runtime is still larger than the one of QGPN-S for most of the examples.

In contrast to our method, we did not implement SNF with a limited memory BFGS-update. The low number of matrix-vector-products given in Table 1 recommends that this would not yield a significantly better performance.

Comparing FISTA with GPN-F and QGPN-F, where FISTA is used to solve the subproblems, we see that GPN-F is not competitive for the mentioned reasons, whereas QGPN-F is far better than FISTA on its own. A similar observation is true for the comparison of SNF with GPN-S and QGPN-S, where the GPN method is still the slowest method but not significantly. Thus, the globalized proximal Newton-type method with limited memory BFGS-update for the Hessian accelerates the performance of the underlying subproblem solver.

5.2 Student's t-Regression with ℓ_1-Penalty

In many applications of inverse problems, the aim is to find a sparse solution $x^* \in \mathbb{R}^n$ of the problem $Ax = b$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Often, b is not known exactly but only a perturbed vector $\hat{b}$. A widespread solution is to consider the penalized problem

$$\min_x \frac{1}{2}\|Ax - \hat{b}\|_2^2 + \lambda \|x\|_1$$

for some $\lambda > 0$. This works well if we have Gaussian errors in the entries of $\hat{b}$. Particularly, the influence of large errors is large. In problems, where the influence of large errors should be weighted less, but the influence of errors in a specific domain should be weighted

method	iter	Newton-iter	sub-iter	function eval	proximity eval	matrix-vector products
GPN-F	11.7	11.7	994	12.7	1 016	3 923
GPN-S	16.9	16.9	33.4	18.0	50.8	296.0
QGPN-F	29.1	29.1	2 015	30.2	2 471	58.3
QGPN-S	21.6	21.6	36.2	22.7	58.9	43.3
FISTA	1 269	-	1 466	4 005	1 466	6 544
SpaRSA	133	-	221	223	222	446
PG	1 520	-	-	3 131	1 520	4 642
SNF	15.2	14.0	31.2	15.7	15.4	90.5

Table 1: Averaged values of 100 runs for the example in Section 5.1 with tolerance 10^{-6}. Abbreviations: iter (total number of (outer) iterations), Newton-iter (number of Newton-iterations – only for GPN-variants and SNF), sub-iter (number of inner iterations), function eval (number of evaluations of the function f or its gradient), proximity eval (number of evaluations of the proximity operator), matrix-vector-products (number of evaluations of products $A \cdot x$ or $A^T \cdot x$).

more, it is reasonable to replace the quadratic loss by the student loss. We obtain the problem

$$\min_x \psi(x) := \sum_{i=1}^{m} \phi\big((Ax-b)_i\big) + \lambda\|x\|_1 = \sum_{i=1}^{m} \log\left(1 + \frac{(Ax-b)_i^2}{\nu}\right) + \lambda\|x\|_1, \qquad (22)$$

with $\phi\colon \mathbb{R} \to \mathbb{R}, \phi(u) = \log\left(1 + \frac{u^2}{\nu}\right)$ for some $\nu > 0$. For more information on student's t-distribution, we refer to [1, 26] and references therein. It is easy to see that the derivative of ϕ is still Lipschitz continuous and ϕ is coercive, but not convex. Thus, many state-of-the-art methods are not applicable to this problem.

5.2.1 Algorithmic Details

Subproblem solvers. As seen in Section 5.1, the SNF subproblem solver performed much better than the FISTA subproblem solver. Thus, we use again the semismooth Newton method with filter globalization [26] for the solution of the subproblems, apply at most 10 inner iterations per outer iteration and adapt the tolerance to get more exact solutions, if the current iterate is close to the solution of the main problem. We denote this method by GPN.

Since the problem in this section is nonconvex, the subproblems might be not bounded from below. To circumvent this problem, we also implemented a variant with regularized Hessians. As the second derivative of ϕ is easy to compute and the Hessian of the objective function is of the form $A^T DA$ for some diagonal matrix $D \in \mathbb{R}^{m\times m}$, we replace all diagonal entries d_i of D by the maximum of d_i and a small positive constant. The subproblem solver remains unchanged and we denote this regularized method by GPN+.

Choice of parameters. As above, we set $p = 2.1$, $\rho = 10^{-8}$, $\beta = 0.1$, and $\sigma = 10^{-4}$. In this case, we start with $c_0 = 100$ and again adapt c_k to approximate the Lipschitz constant of the gradient of the smooth part in (22).

Quasi-Newton-Update. In the second of the following test examples we use again a variant of the globalized proximal Newton method, where the Hessian of f is replaced by a limited memory BFGS-update with a memory of 10. We denote the method by QGPN.

5.2.2 State-of-the-art methods

Since problem (22) is nonconvex, most of the methods in Section 5.1 do not apply in this case. We therefore compare our algorithm to the following methods.

PG. The proximal gradient method as described in Algorithm 2.2 has no convexity requirement. Again, we set $\beta = 0.1$, $\sigma = 10^{-4}$, and $H_k = c_k I$, where c_k is initialized with $c_0 = 100$ and adapted to reach a Lipschitz constant of ∇f.

SNF [26]. The semismooth Newton method with filter globalization, as described in 5.1.1, has also a nonconvex variant with additional descent conditions, which are checked for the semismooth Newton update. We choose all constants as described in [26].

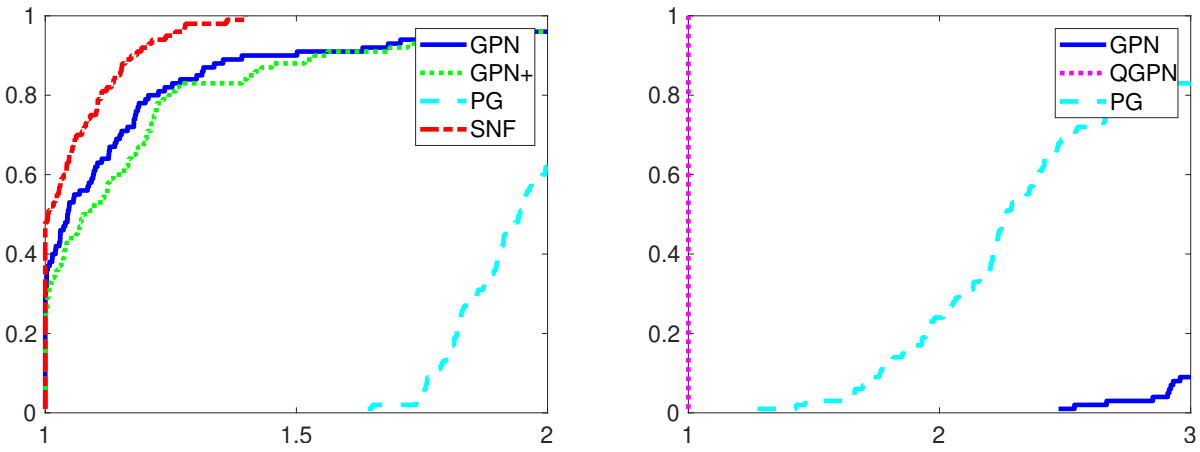
5.2.3 Numerical Comparison

As mentioned above, we test two sets of examples. We start with the test setting described in [26]. Let $n = 512^2$ and $m = n/8 = 32768$. The matrix $A \in \mathbb{R}^{m \times n}$ takes m random cosine measurements, i.e. for a random subset $I \subset \{1, \dots, n\}$ with m elements, we set $Ax = (\mathrm{dct}(c))_I$, where dct is the discrete cosine transform.

We generate a true sparse vector $x^{\mathrm{true}} \in \mathbb{R}^n$ with $k = \lfloor n/40 \rfloor = 6553$ nonzero entries, whose indices are chosen randomly. The nonzero components are computed via $x_i^{\mathrm{true}} = \eta_1(i) 10^{\eta_2(i)}$ with $\eta_1(i) \in \{\pm 1\}$ is a random sign and $\eta_2(i)$ is chosen independently from a uniform distribution in $[0, 1]$. The image $b \in \mathbb{R}^m$ is generated by adding Student's t-noise with degree of freedom 4 and rescaled by 0.1 to Ax^{true}. We set $\nu = 0.25$ and set $\lambda = 0.1\lambda_{\max}$, where $\lambda_{\max}$ is the critical value, for which the zero vector is already a critical point of (22). Using Fermat's rule for the generalized Jacobian of (22), we obtain by a short calculation $\lambda_{\max} = 2 \left\| \sum_{i=1}^m b_i/(\nu + b_i^2) \cdot a_i \right\|_\infty$, where a_i^T is the i-th row of A.

We start with the initial point $x^0 = A^T b$ and, again, terminate each of the algorithms above, when the value $\psi(x^k)$ in the current iterate x^k satisfies (21) for $\mathtt{tol} = 10^{-6}$, where ψ^* is computed by GPN with a very high accuracy. It is important to mention that all stationary points of problem (22), if there is more than one, have the same function value. Thus, this termination criterion makes sense although the problem is nonconvex.

For this example, we do not use QGPN since test runs have shown that QGPN is significantly slower than GPN here. The reason is that, in contrast to the example in 5.1.3, the computation of matrix-vector-products involving the matrix A are cheaper than the product with the BFGS-matrix, as the discrete cosine transform is a predefined Matlab-function.



(a) Example 1 (b) Example 2

Figure 2: Performance profiles showing the runtime for 100 random test examples described in Section 5.2. Figures (a) and (b) correspond to Examples 1 and 2, respectively.

method	iter	Newton-iter	sub-iter	function eval	proximity eval	matrix-vector products
GPN	11.5	11.4	57.4	13.6	76.2	1 475
GPN+	11.6	11.5	58.0	13.7	77.2	1 530
PG	460	-	-	956	460	1 417
SNF	51.0	21.0	231	96.4	66.0	532

Table 2: Averaged values of 100 runs for the first example in Section 5.2 with tolerance 10^{-6}. The columns have the same meaning as in Table 1.

To accomplish comparability, we look at the runtime of 100 test examples and document the performance using the performance profiles introduced by Dolan and Moré [14]. The results are shown in Figure 2 (a), the averaged values for some counters are given in Table 2.

The first observation is that there is no significant difference between the globalized proximal Newton method GPN and the regularized version GPN+. In both methods, almost all updates are performed by proximal Newton steps. Thus, in the following we refer only to GPN.

The proximal gradient method is in all examples significantly slower than the second order methods. As mentioned above, this is not due to the number of matrix-vector-products, which has the same magnitude as the one for GPN. In contrast, the numbers of function evaluations and evaluations of the proximity operator are much higher.

To demonstrate the performance of the limited memory BFGS proximal Newton-type method QGPN, we construct a second test example with higher computation costs for the matrix-vector-products with the matrices A or A^T. In the above test setting, we change n, m and use A as defined in Section 5.1, this is $n = 10^4$, $m = 10^6$, and $A \in \mathbb{R}^{m \times n}$ with approximately 10 nonzero entries in every row. Everything else remains unchained.

method	iter	Newton-iter	sub-iter	function eval	proximity eval	matrix-vector products
GPN	49.2	49.2	246	83.3	330	2 169
GPN+	29.6	29.6	148	68.1	184	3 547
QGPN	125	125	837	211	994	336
PG	156	-	-	572	156	728
SNF	DNC	DNC	DNC	DNC	DNC	DNC

Table 3: Averaged values of 100 runs for the second example in Section 5.2. The columns have the same meaning as in Table 1. The abbreviation DNC stands for: did not converge within 1 000 iterations.

As there was no significant difference in the performance of GPN and GPN+, we apply GPN, QGPN, SNF and the proximal gradient method PG to this setting. The results are shown in Figure 2 (b) and Table 3.

First, we observe that SNF did not converge at all within 1 000 iterations for this problem class. A look at the function value shows that it increases in every step. Since SNF is not a descent method regarding the function value and there is no result guaranteeing the convergence in the nonconvex case, this is not unreasonable.

Comparing the remaining methods, we find that the results confirm the observations of the example in Section 5.1. The performance of QGPN is far the best, whereas GPN is not competitive, though it is not as bad as for the ℓ_1-regularized logistic regression.

5.3 Logistic Regression with Overlapping Group Penalty

The main advantage of the globalized proximal Newton method over semismooth Newton methods is that it is also able to solve problems of type (1), where the nonsmooth function φ is not the ℓ_1-norm and there is no known formula to compute the proximity operator to this function. An example is the group penalty function

$$\varphi(x) = \lambda \sum_{j=1}^{s} \mu_j \|x_{G_j}\|_2,$$

where $\mu_j > 0$ are positive weights, $\lambda > 0$ and $G_j \subset \{1, \dots, n\}$ are nonempty sets. When the sets G_j $(j = 1, \dots, s)$ form a partition of $\{1, \dots, n\}$ or are at least pairwise disjoint, the proximity operator can be computed explicitly. Here we are interested in the case of overlapping groups, i.e. the sets G_j are not pairwise disjoint. In this case, no explicit formula for the proximity operator is known.

Like in Section 5.1 we consider a logistic regression problem

$$\min_x \frac{1}{m} \sum_{i=1}^{m} \phi\big((Ax)_i\big) + \lambda \sum_{j=1}^{s} \mu_j \|x_{G_j}\|_2, \tag{23}$$

where $A \in \mathbb{R}^{m \times n}$ contains the information on feature vectors and corresponding labels and $\phi : \mathbb{R} \to \mathbb{R}$ is defined by $\phi(u) := \log\big(1 + \exp(-u)\big)$. A group penalty makes sense in

many applications here, since some features are related to others. For more information on logistic regression with group penalty, we refer to [24].

5.3.1 Algorithmic Details

Subproblem solver. As there is no formula to compute the proximity operator of φ, the subproblem solvers of the previous sections are not directly applicable. We can write φ as $\tilde{\varphi} \circ B$, where B is a linear mapping and $\tilde{\varphi}$ is a group penalty without overlapping. Thus, we can compute the proximity operator of $\tilde{\varphi}$. Both, the proximal Newton subproblem as well as the proximity operator, can be written as

$$\min_x \frac{1}{2} x^T Q x + c^T x + \tilde{\varphi}(Bx)$$

with a positive definite matrix $Q \in \mathbb{R}^{n \times n}$ and $c \in \mathbb{R}^n$. We solve both problems with fixed point methods described by Chen et al. in [10]. For the computation of the proximity operator, we use the fixed point algorithm based on the proximity operator (FP^{2}O) and for solving the proximal Newton subproblem the primal-dual fixed point algorithm based on the proximity operator (PDFP^{2}O).

For both methods, we use a stopping tolerance of 10^{-9} and apply at most 10 iterations for each problem. For the method we also need the largest eigenvalue of BB^T, which can be shown to be equal to the largest integer k such that there exists an index $i \in \{1, \dots, n\}$ that is contained in k groups G_j.

Choice of parameters. As before, we set the parameters to $p = 2.1$, $\rho = 10^{-8}$, $\beta = 0.1$, and $\sigma = 10^{-4}$. Here, we start with $c_0 = 1$ and again adapt c_k to approximate the Lipschitz constant of the gradient of the smooth part in (23).

Other methods. We make a comparison between our method with the above mentioned subproblem-solvers, FISTA [5] with the parameters as in 5.1.1. For the computation of the proximity operators, we also use FP^{2}O. Furthermore, we apply PDFP^{2}O directly to problem (23).

5.3.2 Numerical Comparison

We follow an example in [2] and generate $A \in \mathbb{R}^{n \times m}$ with $n = 1000$, $m = 700$ from a uniform distribution and normalize the columns of A. The groups G_j are

$$\{1, \dots, 5\},\ \{5, \dots, 9\},\ \{9 \dots, 13\},\ \{13, \dots, 17\},\ \{17, \dots, 21\},$$
$$\{4, 22, \dots, 30\},\ \{8, 31, \dots, 40\},\ \{12, 41, \dots, 50\},\ \{16, 51, \dots, 60\},\ \{20, 61, \dots, 70\},$$
$$\{71, \dots, 80\},\ \{81, \dots, 90\},\ \dots,\ \{991, \dots 1000\}.$$

The first five groups contain five consecutive numbers and the last element of one group is, at the same time, the first element of the next group. Each of the next five groups contain one element of one of the first groups. The remaining groups have no overlap

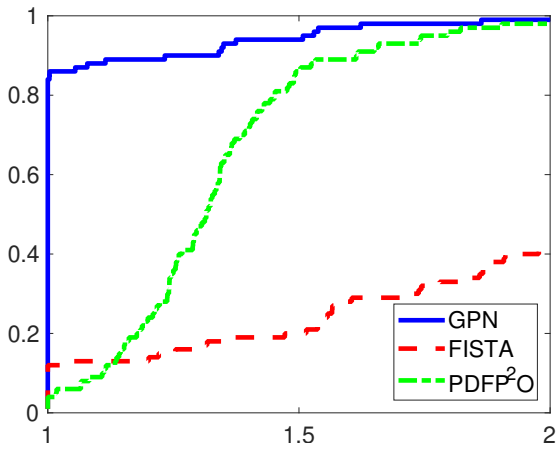


Figure 3: Performance profile showing the runtime for 100 random test examples from Section 5.3 with tolerance 10^{-6}.

method	iter	Newton-iter	sub-iter	matrix-vector products
GPN	9.5	9.5	95.1	221
PDFP^{2}O	76.9	-	-	156
FISTA	23.4	-	234	119

Table 4: Averaged values of 100 runs for the example in Section 5.3 using the tolerance 10^{-6} and three different methods.

and contain always 10 elements. The coefficients μ_j are chosen to be $1/\sqrt{|G_j|}$, where $|G_j|$ is the number of indices in that group.

The parameter λ is again chosen as $0.1\lambda_{\max}$, where $\lambda_{\max}$ is the critical value such that 0 is a solution of (23) for all $\lambda \geq \lambda_{\max}$. Let a_i^T be the rows of A. Then a short computation shows $\lambda_{\max} = \sqrt{5}/(2m) \left\| \sum_{i=1}^m a_i \right\|_2$. As before, we start with the initial value $x^0 = 0$.

We terminate each of the algorithm as soon as the current iterate satisfies (21) for $\texttt{tol} = 10^{-6}$, where ψ^* is the function value computed by GPN using a very high accuracy. Again, we document the results using the performance profiles on the runtime of 100 test examples. The results are shown in Figure 3, the averaged values for some counters are given in Table 4.

We see that there are about 15% of the examples, where FISTA performs better than GPN, but in most examples GPN shows by far the best performance. This can be seen by looking at the number of inner iterations of both methods. In this case, the costs of inner iterations is almost equal for both methods. Since the average number of inner iterations in FISTA is more then twice as big as the one of GPN, this illustrates the difference in performance.

6 Conclusion

We introduced a globalization of the proximal Newton-type method to solve structured optimization problems consisting of a smooth and a convex function. For this purpose the proximal Newton-type method was combined with a proximal gradient method using a novel descent criterion. We also gave an inexactness approach and the possibility to replace the Hessian of the smooth part by quasi-Newton matrices. We proved global convergence in the convex and nonconvex case and, under suitable conditions, local superlinear convergence.

The numerical part shows that the proposed method is competitive for convex and nonconvex problems, especially when the computation of the Hessian is expensive and we can use limited memory quasi-Newton updates. Furthermore, when there is no efficient way to compute the proximity operator for the nonsmooth function, the globalized proximal Newton-type method outperforms the methods compared to.

References

[1] A. Aravkin, M. P. Friedlander, F. J. Herrmann, and T. Van Leeuwen, *Robust inversion, dimensionality reduction, and randomized sampling*, Mathematical Programming, 134 (2012), pp. 101–125.

[2] A. Argyriou, C. A. Micchelli, M. Pontil, L. Shen, and Y. Xu, *Efficient first order methods for linear composite regularizers*, arXiv preprint arXiv:1104.1436, (2011).

[3] H. Bauschke and P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books in Mathematics, Springer International Publishing, 2 ed., 2017.

[4] A. Beck, *First-Order Methods in Optimization*, MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, 2017.

[5] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends in Machine Learning, 3 (2011), pp. 1–122.

[7] R. H. Byrd, J. Nocedal, and F. Oztoprak, *An inexact successive quadratic approximation method for l-1 regularized optimization*, Mathematical Programming, 157 (2016), pp. 375–396.

[8] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi-Newton matrices and their use in limited memory methods*, Mathematical Programming, 63 (1994), pp. 129–156.

[9] D.-Q. Chen, Y. Zhou, and L.-J. Song, *Fixed point algorithm based on adapted metric method for convex minimization problem with application to image deblurring*, Advances in Computational Mathematics, 42 (2016), pp. 1287–1310.

[10] P. Chen, J. Huang, and X. Zhang, *A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration*, Inverse Problems, 29, pp. 025011, 33.

[11] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Modeling & Simulation, 4 (2005), pp. 1168–1200.

[12] T. De Luca, F. Facchinei, and C. Kanzow, *A semismooth equation approach to the solution of nonlinear complementarity problems*, Mathematical Programming, 75 (1996), pp. 407–439.

[13] J. E. Dennis and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Mathematics of Computation, 28 (1974), pp. 549–560.

[14] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical programming, 91 (2002), pp. 201–213.

[15] K. Fountoulakis and R. Tappenden, *A flexible coordinate descent method*, Computational Optimization and Applications, 70 (2018), pp. 351–394.

[16] M. Fukushima and H. Mine, *A generalized proximal point algorithm for certain non-convex minimization problems*, International Journal of Systems Science, 12 (1981), pp. 989–1000.

[17] H. Ghanbari and K. Scheinberg, *Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates*, Computational Optimization and Applications, 69 (2018), pp. 597–627.

[18] B. Gu, Z. Huo, and H. Huang, *Inexact proximal gradient methods for non-convex and non-smooth optimization*, arXiv preprint arXiv:1612.06003, (2016).

[19] K. Koh, S.-J. Kim, and S. Boyd, *An interior-point method for large-scale l1-regularized logistic regression*, Journal of Machine Learning Research, 8 (2007), pp. 1519–1555.

[20] C.-p. Lee and S. J. Wright, *Inexact successive quadratic approximation for regularized optimization*, Comput. Optim. Appl., 72 (2019), pp. 641–674.

[21] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM Journal on Optimization, 24 (2014), pp. 1420–1443.

[22] J. Li, M. S. Andersen, and L. Vandenberghe, *Inexact proximal Newton methods for self-concordant functions*, Mathematical Methods of Operations Research, (2016), pp. 1–23.

[23] Q. Li, L. Shen, Y. Xu, and N. Zhang, *Multi-step fixed-point proximity algorithms for solving a class of optimization problems arising from image processing*, Advances in Computational Mathematics, 41 (2015), pp. 387–422.

[24] L. Meier, S. Van De Geer, and P. Bühlmann, *The group lasso for logistic regression*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70 (2008), pp. 53–71.

[25] A. Milzarek, *Numerical methods and second order theory for nonsmooth problems*, PhD thesis, Technische Universität München, 2016.

[26] A. Milzarek and M. Ulbrich, *A semismooth Newton method with multidimensional filter globalization for l_1-optimization*, SIAM Journal on Optimization, 24 (2014), pp. 298–333.

[27] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572.

[28] J.-J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bulletin de la Société mathématique de France, 93 (1965), pp. 273–299.

[29] Y. Nesterov, *Gradient methods for minimizing composite functions*, Mathematical Programming, 140 (2013), pp. 125–161.

[30] P. Patrinos and A. Bemporad, *Proximal Newton methods for convex composite optimization*, in 52nd IEEE Conference on Decision and Control, IEEE, 2013, pp. 2358–2363.

[31] P. Patrinos, L. Stella, and A. Bemporad, *Forward-backward truncated Newton methods for convex composite optimization*, arXiv preprint arXiv:1402.6655, (2014).

[32] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[33] K. Scheinberg, D. Goldfarb, and X. Bai, *Fast first-order methods for composite convex optimization with backtracking*, Foundations of Computational Mathematics, 14 (2014), pp. 389–417.

[34] K. Scheinberg and X. Tang, *Practical inexact proximal quasi-Newton method with global complexity analysis*, Mathematical Programming, 160 (2016), pp. 495–529.

[35] L. Stella, A. Themelis, and P. Patrinos, *Forward–backward quasi-Newton methods for nonsmooth optimization problems*, Computational Optimization and Applications, 67 (2017), pp. 443–487.

[36] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher, *A proximal Newton framework for composite minimization: Graph learning without Cholesky decompositions and matrix inversions*, in International Conference on Machine Learning, 2013, pp. 271–279.

[37] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Mathematical Programming, 117 (2009), pp. 387–423.

[38] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Transactions on Signal Processing, 57 (2009), pp. 2479–2493.

[39] M.-C. Yue, Z. Zhou, and A. M.-C. So, *A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo–Tseng error bound property*, Mathematical Programming, 174 (2019), pp. 327–358.

[40] S. Zhang, H. Qian, and X. Gong, *An alternating proximal splitting method with global convergence for nonconvex structured sparsity optimization*, in 30. AAAI Conference on Artificial Intelligence, 2016, pp. 2330–2336.