

Projected Subgradient Methods for a Class of Nonsmooth and Nonconvex Optimization Problems

Christian Kanzow* Jannis Krüger[†] Leo Lehmann[‡]

Wednesday 2nd September, 2026

Abstract

We investigate the optimization problem of minimizing a nonsmooth function that satisfies a nonsmooth version of the descent lemma over a nonempty and closed but not necessarily convex set. The objective function belongs to the class of upper- $\mathcal{C}^2$ functions, whereas the constraints may promote a sparse or low-rank structure. We propose a projected subgradient method with two different globalization strategies: (a) a nonmonotone linesearch and, under additional assumptions, (b) an auto-conditioned method, where the stepsize is given by a formula depending on data from past iterations. We show that both methods converge to solutions that satisfy a stronger stationarity concept than one would expect from the subdifferential sum-rule, which is particularly important since the optimization problems of interest are inherently nonconvex. Finally, we present promising numerical results when applying the algorithm to an MPEC-style problem as well as the matrix optimization problems MAXCUT and Robust PCA.

Keywords. nonsmooth optimization, nonconvex optimization, projected subgradient method, nonmonotone line search, line search free methods, stationary point, critical point, tangent and normal cones

AMS subject classifications. 90C30, 49J53, 65K05

1 Introduction

Let us consider the optimization problem

$$\min_{x \in D} \varphi(x), \tag{1.1}$$

where $D \subseteq \mathbb{X}$ is a nonempty and closed but not necessarily convex subset of some Euclidean space $\mathbb{X}$. We assume that the objective function φ satisfies a nonsmooth (and local) formulation of the descent lemma. The corresponding class of functions is called upper- $\mathcal{C}^2$ and has been

*University of Würzburg, Institute of Mathematics, Emil-Fischer-Str. 30, 97074 Würzburg, Germany; e-mail: christian.kanzow@uni-wuerzburg.de

[†]University of Würzburg, Institute of Mathematics, Emil-Fischer-Str. 30, 97074 Würzburg, Germany; e-mail: jannis.krueger@uni-wuerzburg.de

[‡]University of Würzburg, Institute of Mathematics, Emil-Fischer-Str. 30, 97074 Würzburg, Germany; e-mail: leo.lehmann@uni-wuerzburg.de

shown to have favorable properties for the application of subgradient methods with linesearch [1]. Notably, every differentiable function with locally Lipschitz continuous gradient is upper- $\mathcal{C}^2$. On the other hand, every concave function is also upper- $\mathcal{C}^2$. Moreover, the sum of upper- $\mathcal{C}^2$ functions is still an upper- $\mathcal{C}^2$ function. Consequently, many DC (DC = difference-of-convex) programs belong to the class of problem (1.1). This motivates our consideration of constrained optimization problems as in (1.1).

In the differentiable case, it has recently been shown in [20] that projected gradient methods with linesearch accumulate at Bouligand stationary points. In the case of a local Lipschitz gradient, the accumulation points are even proximally stationary, meaning that the negative gradient is a so-called proximal normal. Notably, the proximal normal cone is in general smaller than both the regular and limiting normal cone. We show that by applying a projected subgradient method with (nonmonotone) linesearch to our problem (1.1), one obtains that limits of the subgradient sequence are contained in the proximal normal cone. This is a stronger stationarity than that obtained directly from calculus rules of the limiting subdifferential as the latter only guarantees the existence of a negative subgradient that is a limiting normal to D .

For a continuously differentiable function φ , the projected gradient method is a standard method for optimization over a nonempty closed set D . It is based on the iteration

$$x^{k+1} \in \Pi_D(x^k - \tau_k \nabla \varphi(x^k)),$$

where τ_k is a stepsize and Π_D is the possibly set-valued projection onto D . In our descent method, where we allow for nonsmooth objective functions φ , we replace the gradient by an arbitrary element $w^k \in \partial\varphi(x^k)$, where $\partial\varphi$ denotes the limiting (or Clarke) subdifferential of φ .

The choice of τ_k is crucial for the convergence behavior of the algorithm. Using a backtracking linesearch, the stepsize is chosen such that a suitable linesearch-criterion holds. However, by using the Armijo condition (see [2]) a descent in the function value is enforced in every iteration. In order to obtain (possibly) larger stepsizes and fewer backtracking iterations, we employ a nonmonotone method [1, 4, 8]. In the literature, there are essentially two approaches to obtain nonmonotone stepsize rules: The max-type rule by Grippo et al. [12] and the mean-type rule by Zhang and Hager [27]. In the constrained case, both have been studied for differentiable objective functions in [20]. On the other hand, without the projection step, they were also employed for our class of nonsmooth objectives in [1] and [15], respectively.

As a second contribution, under the stronger assumption that the nonsmooth formulation of the descent lemma holds with a global constant, we propose an auto-conditioned method, where the stepsize is calculated from an explicit formula depending on past iterates, function values and subgradients. This line of work has recently attracted a lot of interest as it may lead to a significant reduction in per-iteration cost compared the linesearch methods and still needs no problem-dependent parameters as input. A similar approach was proposed in the pioneering work [17] for convex and differentiable objectives, whereas our stepsize is closely related to the works [16] and [26].

The paper is organized in the following way. In Section 2, we introduce some notation and concepts from variational analysis. We define the class of upper- $\mathcal{C}^2$ functions and recall some of their properties that make them compelling for optimization methods. Section 3 presents our nonmonotone descent method to solve (1.1). We discuss its properties and give (global) convergence guarantees. The subsequent Section 4 introduces the auto-conditioned method for (1.1). Again, we obtain essentially the same convergence results, however, under additional and stronger assumptions. Next, in Section 5, we apply the algorithm to do numerical experiments on three theoretically and practically relevant examples. Lastly, we conclude with some remarks in Section 6.

Notation: We will consider optimization problems on some nonempty closed set $D \subseteq \mathbb{X}$, where $\mathbb{X}$ is a Euclidean space (finite-dimensional Hilbert space). We identify the dual space $\mathbb{X}^*$ with $\mathbb{X}$ itself. We write $\langle x, y \rangle$ for the scalar product of two elements $x, y \in \mathbb{X}$ and $\|x\|$ for the

induced norm of some $x \in \mathbb{X}$. The projection of $x \in \mathbb{X}$ onto D is denoted by $\Pi_D(x) := \{y \in \mathbb{X} \mid \text{dist}(x, D) = \|x - y\|\}$, where the distance function is $\text{dist}(x, D) := \inf_{y \in D} \|x - y\|$. If D is, in addition, convex, $\Pi_S(x)$ is single-valued for all $x \in \mathbb{X}$. Also, we write the indicator function of D as δ_D , which is 0 for arguments in D and ∞ else.

We denote by $B_r(x)$ the open and $\overline{B}_r(x)$ the closed ball of some radius $r > 0$ around $x \in \mathbb{X}$. For arbitrary $S \subseteq \mathbb{X}$, we denote its interior by $\text{int}(S)$, the closure by $\text{cl}(S)$, and the convex hull by $\text{conv}(S)$.

Finally, let $\mathbb{R}$ denote the set of real numbers, while $\overline{\mathbb{R}} := (-\infty, +\infty]$ is the set of extended reals except that we exclude the value $-\infty$. Given an extended-valued function $\theta : \mathbb{X} \rightarrow \overline{\mathbb{R}}$, we call $\text{dom}(\theta) := \{x \in \mathbb{X} \mid \theta(x) < \infty\}$ the domain of θ . The function θ is said to be proper if $\text{dom}(\theta)$ is nonempty.

2 Background Material

Let us recall some results from variational analysis. We refer the interested reader to the two excellent monographs [18, 21] for more details.

Given a proper, lower semicontinuous function $g : \mathbb{X} \rightarrow \overline{\mathbb{R}}$ and any $x \in \text{dom}(g)$, we call

$$\partial^F g(x) := \left\{ v \in \mathbb{X} \mid \liminf_{y \rightarrow x, y \neq x} \frac{g(y) - g(x) - \langle v, y - x \rangle}{\|y - x\|} \geq 0 \right\}$$

the *regular* or *Fréchet subdifferential* of f at x , whereas

$$\partial^M g(x) := \{v \in \mathbb{X} \mid \exists x^k, v^k \in \mathbb{X} : x^k \rightarrow x, g(x^k) \rightarrow g(x), v^k \rightarrow v, v^k \in \partial^F g(x^k) \forall k\}$$

is called the *limiting*, *Mordukhovich*, or *basic subdifferential* of g at x .

For a function g that is locally Lipschitz continuous around $\bar{x} \in \mathbb{R}$, we define the *Clarke subdifferential* of g at $\bar{x}$ as

$$\partial^C g(\bar{x}) := \{v \in \mathbb{X} : \langle v, h \rangle \leq g^\circ(\bar{x}; h) \text{ for all } h \in \mathbb{X}\},$$

where

$$g^\circ(\bar{x}; h) := \limsup_{x \rightarrow \bar{x}, t \rightarrow 0^+} \frac{g(x + th) - g(x)}{t}$$

is the *Clarke directional derivative* of g at $\bar{x}$ in the direction h . Later, for results that hold true for both the limiting and the Clarke subdifferential, we use the following convention:

$$\partial g \text{ means either } \partial^M g \text{ or } \partial^C g.$$

Let us recall the following properties: It holds that $\partial^C g(\bar{x}) = \text{conv}(\partial^M g(\bar{x}))$ for all locally Lipschitz continuous functions. Further, in that case, the Clarke subdifferentials of g on bounded subsets of $\mathbb{X}$ remain bounded, too, see e.g. [7]. By consequence, the same is true for the limiting subdifferential. Both the Clarke and the limiting subdifferential share the following so-called *robustness* property: Assume that $\{x^k\}$ is a sequence converging to some limit $\bar{x}$, and $w^k \in \partial g(x^k)$ for all $k \in \mathbb{N}$ converges to some $\bar{w} \in \mathbb{X}$. Then it holds that $\bar{w} \in \partial g(\bar{x})$.

Let us next introduce the class of upper- $\mathcal{C}^2$ functions. These are central for our method and the convergence analysis thereof. For more details on this class of functions we refer to [21].

Definition 2.1. Let $U \subseteq \mathbb{X}$ be an open set. We say that a function $\varphi : U \rightarrow \mathbb{R}$ is upper- $\mathcal{C}^2$ on U , if, on some neighborhood V of each $\bar{x} \in U$, there is a representation

$$\varphi(x) = \min_{c \in C} \varphi_c(x),$$

where the functions φ_c are of class $\mathcal{C}^2$ on V , and C is a compact set (in some topological space) such that φ_c and its first- and second-order partial derivatives depend continuously on $(x, c) \in V \times C$.

For example, taking the discrete topology, it follows that functions that are the pointwise minimum of a finite family of twice continuously differentiable functions $f_i : U \rightarrow \mathbb{R}$ on some open set $U \subseteq \mathbb{X}$ are upper- $\mathcal{C}^2$. Further examples can be derived from the subsequent characterization of upper- $\mathcal{C}^2$ functions from the recent report [1, Prop. 3.2]. We also refer to the later Section 5 for applications where these functions arise.

Proposition 2.2. *Let $U \subseteq \mathbb{X}$ be an open set and $\varphi : \mathbb{X} \rightarrow \mathbb{R}$ be locally Lipschitz on U . Then the following statements are equivalent:*

(a) φ is upper- $\mathcal{C}^2$ on U .

(b) For each $\bar{x} \in U$, there exist a constant $\kappa \geq 0$ and some neighborhood V of $\bar{x}$ such that

$$\varphi(y) \leq \varphi(x) + \langle w, y - x \rangle + \kappa \|y - x\|^2 \quad (2.1)$$

for all $x, y \in V$ and all $w \in \partial\varphi(x)$.

(c) For each $\bar{x} \in U$, there exists some neighborhood V of $\bar{x}$ where φ can be expressed as $\varphi = g - h$, where g is differentiable with Lipschitz gradient, and h is Lipschitz and prox-regular (see [21, Definition 3.27]). Indeed, one can take $g = \kappa \|\cdot\|^2$, for some $\kappa \geq 0$, and h to be convex.

We will later see that inequality (2.1) from part (b) is crucial for our convergence analysis. Namely, it is a nonsmooth counterpart of the usual descent lemma for smooth functions with Lipschitz gradient. In this analogy, the constant 2κ in (2.1) corresponds to the Lipschitz constant. Note, however, that in the inequality above, κ is a local constant possibly depending on the given point $\bar{x}$. Inequality (2.1) holds for an arbitrary subgradient $w \in \partial\varphi(x)$ and not just for a particular element from $\partial\varphi(x)$. This is a very strong observation that is later important to show that our subgradient method is well-defined.

Finally, the characterization in (c) links upper- $\mathcal{C}^2$ functions to the class of DC-functions, see the corresponding discussion in [1]. More precisely, DC-functions where the first term is smooth are upper- $\mathcal{C}^2$.

We proceed by showing that if φ is upper- $\mathcal{C}^2$ on U and $D \subset U$ is a compact subset, then the constant κ in characterization (b) can be chosen as a global constant. This result and its proof are essentially the same as for the corresponding fact that every locally Lipschitz continuous function is globally Lipschitz on compact sets.

Lemma 2.3. *Let $\varphi : \mathbb{X} \rightarrow \mathbb{R}$ be upper- $\mathcal{C}^2$ on an open set $U \subseteq \mathbb{X}$ and let $C \subset U$ be compact. Then there exists a constant κ_C such that*

$$\varphi(y) \leq \varphi(x) + \langle w, y - x \rangle + \kappa_C \|y - x\|^2 \quad (2.2)$$

holds for all $x, y \in C$ and all $w \in \partial\varphi(x)$.

Proof. As φ is locally Lipschitz continuous, $\partial\varphi$ is locally bounded, hence, on the compact set C , there exists a constant G satisfying $\|w\| \leq G$ for all $w \in \partial\varphi(x)$, for all $x \in C$.

As φ is upper- $\mathcal{C}^2$, for every $\bar{x} \in C$, we find a constant $\kappa_{\bar{x}}$ and an open neighborhood $V_{\bar{x}}$, on which (2.1) holds (with $\kappa = \kappa_{\bar{x}}$). Without loss of generality, we can assume $V_{\bar{x}} = B_{r_{\bar{x}}}(\bar{x})$. Now, $\{B_{r_{\bar{x}}}(\bar{x})\}_{\bar{x} \in C}$ is an open cover of C , and by compactness, we find a finite open subcover, $\{B_{r_i}(\bar{x}_i)\}_{i=1}^n$. Now, define

$$M := \sup_{x, y \in C, w \in \partial\varphi(x)} \{\varphi(y) - \varphi(x) - \langle w, y - x \rangle\},$$

which is finite due to $\|w\| \leq G$ for all $w \in \partial\varphi(x)$, for all $x \in C$ and continuity in x and y . Let $\kappa := \max\{\kappa_{\bar{x}_1}, \dots, \kappa_{\bar{x}_n}, \frac{M}{r^2}\}$, where $r > 0$ is Lebesgue's number associated with the subcover and we assume $r < r_i$ for $i = 1, \dots, n$.

Let now $x, y \in C$ be arbitrary. Then there are two cases: First, assume $\|x - y\| \geq r$, then

$$\varphi(y) - \varphi(x) - \langle w, y - x \rangle \leq M \leq \frac{M}{r^2} \|x - y\|^2 \leq \kappa \|x - y\|^2.$$

Secondly, if $\|x - y\| < r$, then, by definition of Lebesgue's number, there exists some $i \in \{1, \dots, n\}$, such that $x, y \in B_{r_i}(\bar{x}_i)$, hence

$$\varphi(y) - \varphi(x) - \langle w, y - x \rangle \leq \kappa_{\bar{x}_i} \|x - y\|^2 \leq \kappa \|x - y\|^2.$$

This completes the proof. $\square$

If C is a compact set, then we denote by κ_C the constant such that (2.2) holds. For our auto-conditioned method later, we note that there are other cases where κ in part (b) of Proposition 2.2 can be chosen as a global constant. For example, assume that φ has a global representation as a minimum over a compact set, i.e. $\varphi(x) = \min_{c \in C} \varphi_c(x)$ for all $x \in U$ ($U \subseteq \mathbb{X}$ open) with φ_c satisfying the properties as in Definition 2.1. Then, by [21, Theorem 10.33], if ρ is an upper bound to the Hessians $\|\nabla^2 \varphi_c(x)\|$ over $C \times U$, then φ is upper- $\mathcal{C}^2$ with the (global) constant $\kappa = 2\rho$.

The projection mapping has the following important properties that we cite from [20].

Proposition 2.4. *Let $D \subset \mathbb{X}$ be nonempty and closed. Then for all $x \in D$, $v \in \mathbb{X}$, $y \in \Pi_D(x - v)$, it holds that*

$$\|y - x\| \leq 2\|v\|, \tag{2.3}$$

$$2\langle v, y - x \rangle \leq -\|y - x\|^2, \tag{2.4}$$

with strict inequalities in the case that $x \notin \Pi_D(x - v)$.

Let us also recall some notation from variational geometry from [21]. In the following $D \subset \mathbb{X}$ is assumed to be closed and nonempty. We define the *tangent cone* of D at $\bar{x} \in D$ as

$$T_D(\bar{x}) := \left\{ d \in \mathbb{X} \mid \exists \{x^k\} \subset D, t_k \searrow 0, x^k \rightarrow \bar{x}, \frac{x^k - \bar{x}}{t_k} \rightarrow d \right\}.$$

Further, the *regular (or Fréchet) normal cone* is given by

$$N_D^F(\bar{x}) = T_D(\bar{x})^\circ := \left\{ v \in X \mid \langle v, d \rangle \leq 0 \forall d \in T_D(\bar{x}) \right\}.$$

and the *limiting normal cone* is defined as

$$N_D^M(\bar{x}) = \left\{ v \in \mathbb{X} \mid \exists \{x^k\} \subset C, \{v^k\} : x^k \rightarrow \bar{x}, v^k \in N_D^F(\bar{x}), v^k \rightarrow v \right\}.$$

Lastly, we define $N_D^P(\bar{x})$ as the *proximal normal cone* to D at $\bar{x}$ as follows:

$$N_D^P(\bar{x}) = \left\{ v \in \mathbb{X} \mid \exists \bar{\tau} > 0 : \bar{x} \in \Pi_D(\bar{x} + \bar{\tau}v) \right\}.$$

Note that v being proximal normal to D at x implies $\Pi_D(x + \tau v) = \{x\}$ for all $\tau \in [0, \bar{\tau})$. The proximal normal cone $N_D^P(\bar{x})$ is a convex cone and the following inclusions hold and are in general strict (see, e.g. [20]):

$$N_D^P(\bar{x}) \subseteq N_D^F(\bar{x}) \subseteq N_D^M(\bar{x}). \tag{2.5}$$

Next, we introduce our notion of critical points. Let φ be upper- $\mathcal{C}^2$ and let D be a nonempty, closed set. We call $\bar{x} \in D$ a *proximal critical point*, if it holds that

$$0 \in \partial\varphi(\bar{x}) + N_D^P(\bar{x}). \tag{2.6}$$

Equivalently, there exists some $w^* \in \partial\varphi(\bar{x})$ such that $-w^* \in N_D^P(\bar{x})$.

Below, in Proposition 2.5, we will show that a more restrictive version of criticality is a necessary condition for local minimizers of our optimization problem. Namely, if $\bar{x}$ is a local minimizer for (1.1), then

$$-\partial\varphi(\bar{x}) \subseteq N_D^P(\bar{x}).$$

We note that, if D is convex, both concepts are related to the respective terms in DC programming (see e.g. [23]) and, taking the results from Proposition 2.2 into account, one can establish a link between optimization problems for upper- $\mathcal{C}^2$ functions and DC programs.

Even the notion for critical points from (2.6) here is stronger than usual necessary optimality conditions. In general, as φ is locally Lipschitz continuous, by the sum-rule of the limiting subdifferential, one obtains $\partial(\varphi + \delta_D)(x) \subseteq \partial\varphi(x) + N_D^M(x)$, which together with the optimality condition $0 \in \partial(\varphi + \delta_D)(x)$ guarantees that there exists an element $w \in \partial\varphi(x)$, such that $-w \in N_D^M(x)$. In the upcoming proposition, however, by properties of the projection, the limiting normal cone $N_D^M(x)$ can be replaced by the smaller proximal normal cone $N_D^P(x)$. Secondly, the upper- $\mathcal{C}^2$ property guarantees that every negative subgradient is an element of the normal cone.

Proposition 2.5. *Under our usual assumptions, let $x \in D$ be a local minimum of (1.1). Then for all $w \in \partial\varphi(x)$ it holds that $-w \in N_D^P(x)$.*

Proof. By contradiction, assume that there exists some $w \in \partial\varphi(x)$ such that $-w \notin N_D^P(x)$. Let $\rho \in (0, \infty)$. Then for all $\tau \in \left(0, \frac{\rho}{2\|w\|}\right]$, it holds by (2.3) that

$$x \notin \Pi_D(x - \tau w) \subseteq B_{2\tau\|w\|}(x) \subseteq B_\rho(x).$$

Hence, now for all $\tau \in \left(0, \min\left\{\frac{\rho}{2\|w\|}, \frac{1}{2\kappa_{\overline{B}_\rho(x)}}\right\}\right)$ and $y \in \Pi_D(x - \tau w)$ (which implies $y \in B_\rho(x)$), we have by means of Lemma 2.3 and (2.4) that

$$\varphi(y) - \varphi(x) \leq \langle w, y - x \rangle + \kappa_{\overline{B}_\rho(x)}\|y - x\|^2 < \left(\frac{-1}{2\tau} + \kappa_{\overline{B}_\rho(x)}\right)\|y - x\|^2 \leq 0,$$

which contradicts the fact that x was a local minimum. $\square$

By the inclusions in (2.5), we can also define weaker notions of criticality depending on the normal cone in use. This way, we define

- *F-criticality*: $0 \in \partial\varphi(\bar{x}) + N_D^F(\bar{x})$.
- *M-criticality*: $0 \in \partial\varphi(\bar{x}) + N_D^M(\bar{x})$.
- *C-criticality*: $0 \in \partial\varphi(\bar{x}) + N_D^C(\bar{x})$, where the Clarke normal cone $N_D^C(\bar{x})$ is defined as the closed convex hull of $N_D^M(\bar{x})$.

These definitions are generalizations of the respective stationarity conditions for a smooth objective, cf. [20]. The traditional way to define $*$ -stationarity for $*$ $\in \{F, M, C, P\}$ would be to have

$$0 \in \partial^*(\varphi + \delta_D)(\bar{x}),$$

which is in general stronger than criticality as defined above. In our analysis we obtain results for P-criticality, as we utilize the robustness of the limiting subdifferential applied to the sequence of subgradients $w^k \in \partial\varphi(x^k)$ in our algorithm.

3 Nonmonotone Linesearch Method

The standard projected gradient method for the minimization of a continuously differentiable function φ over a closed nonempty set D is based on the iteration

$$x^{k+1} \in \Pi_D(x^k - \tau_k \nabla \varphi(x^k)), \quad (3.1)$$

with a stepsize τ_k usually chosen by a backtracking linesearch, i.e. by multiplying it with some $\beta \in (0, 1)$ from an initial estimate in $[\tau_{\min}, \tau_{\max}]$ until a suitable linesearch criterion holds. The most common linesearch condition is the Armijo rule

$$\varphi(x^{k+1}) \leq \varphi(x^k) - \sigma \tau_k \|\nabla \varphi(x^k)\|^2 \quad (3.2)$$

for some constant $\sigma \in (0, 1)$. The descent method considered in this section is a direct generalization of this approach to the class of objective functions φ which are upper- $\mathcal{C}^2$. In the monotone version, it is based on the iteration (3.1) for arbitrary elements $w^k \in \partial \varphi(x^k)$ and $y^k \in \Pi_D(x^k - \tau_k w^k)$. The counterpart of the (monotone) Armijo rule (3.2) reads

$$\varphi(y^k) \leq \varphi(x^k) + \sigma \langle w^k, y^k - x^k \rangle, \quad (3.3)$$

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

The following result similar to [1, Proposition 4.3] and [20] shows that this Armijo-type condition is satisfied for all sufficiently small stepsizes provided that φ is indeed upper- $\mathcal{C}^2$. Note that the interval in the definition of $\bar{\rho}$ is non-empty by boundedness of the subdifferential over compact sets.

Proposition 3.1. *Let $\bar{x} \in D$, $\tau_{\max} \in (0, \infty)$, $\sigma \in (0, 1)$ and $\rho \in (0, \infty)$. Define*

$$\bar{\rho} \in \left[\rho + 2\tau \sup_{x \in \bar{B}_\rho(\bar{x}) \cap D, w \in \partial \varphi(x)} \|w\|, \infty \right) \text{ and } \tau^* = \frac{1 - \sigma}{2\kappa_{\bar{B}_\rho(\bar{x}) \cap D}}.$$

Then, for all $x \in \bar{B}_\rho(\bar{x}) \cap D$, $\tau \in [0, \min\{\tau^, \tau_{\max}\}]$, $y \in \Pi_D(x - \tau w)$, $w \in \partial \varphi(x)$, it holds that*

$$\varphi(y) \leq \varphi(x) + \sigma \langle w, y - x \rangle. \quad (3.4)$$

Proof. For all $x \in \bar{B}_\rho(\bar{x}) \cap D$, $\tau \in [0, \tau_{\max}]$, $y \in \Pi_D(x - \tau w)$, $w \in \partial \varphi(x)$, it holds by the triangle inequality and (2.3) that

$$\|y - \bar{x}\| \leq \|y - x\| + \|x - \bar{x}\| \leq 2\tau \|w\| + \rho \leq \bar{\rho},$$

which in turn implies $\Pi_D(x - \tau w) \subseteq \bar{B}_{\bar{\rho}}(\bar{x})$. Consequently, we have for all $x \in \bar{B}_\rho(\bar{x}) \cap D$, $\tau \in [0, \min\{\tau^*, \tau_{\max}\}]$, $y \in \Pi_D(x - \tau w)$, $w \in \partial \varphi(x)$, that by Lemma 2.3 the following inequalities hold:

$$\begin{aligned} \varphi(y) &\leq \varphi(x) + \langle w, y - x \rangle + \kappa_{\bar{B}_{\bar{\rho}}(\bar{x})} \|y - x\|^2 \\ &\leq \varphi(x) + (1 - 2\tau \kappa_{\bar{B}_{\bar{\rho}}(\bar{x})}) \langle w, y - x \rangle \\ &\leq \varphi(x) + \sigma \langle w, y - x \rangle. \end{aligned}$$

The final inequality exploits the fact that $\langle w, y - x \rangle \leq 0$ in view of (2.4). $\square$

Note that in particular, the inner backtracking loop terminates with a stepsize contained in some compact interval

$$\tau \in I := [\min\{\tau_{\min}, \beta \tau^*\}, \tau_{\max}]. \quad (3.5)$$

We next provide a generalization to a nonmonotone version of the previous iteration. From (3.4) we know that an algorithm with the iteration above and τ_k determined by a backtracking

linesearch with termination criterion as in (3.4) will produce a monotonically decreasing sequence of function values $\{\varphi(x^k)\}_{k \in \mathbb{N}}$.

Now, (3.4) will also hold if $\varphi(x)$ is replaced by an arbitrary upper bound. Therefore, we assume that we have reference values $\mathcal{R}_k \geq \varphi(x^k)$. Then Proposition 3.1 guarantees finite termination of a backtracking linesearch with the corresponding termination criterion

$$\varphi(y^k) \leq \mathcal{R}_k + \sigma \langle w^k, y^k - x^k \rangle, \quad (3.6)$$

with $y^k \in \Pi_D(x^k - \tau_k w^k)$. Hence, the linesearch for the stepsize parameter is well-defined for upper- $\mathcal{C}^2$ functions.

Crucially, the reason why nonmonotone methods may outperform their monotone counterparts in applications is that (3.6) may allow for larger steps compared to a monotone method.

Before presenting the algorithmic details, let us discuss possible choices for the reference values $\mathcal{R}_k$. One popular choice is due to Grippo et al. [12], where $\mathcal{R}_k := \max\{\varphi(x^j) \mid j = k, k-1, \dots, k-m_k\}$ for some given (but bounded) sequence $m_k \in \mathbb{N}$. As $\mathcal{R}_k$ is the maximum function value over the last few $m_k + 1$ iterates, we refer to this strategy as the *max-rule*. The max-rule for unconstrained minimization of upper- $\mathcal{C}^2$ functions was studied in [1]. There, stationarity of accumulation points was derived.

In our Algorithm 3.2, we employ the reference values as introduced by Zhang and Hager [27]. Here, $\mathcal{R}_{k+1}$ is computed as a convex combination of the previous reference value $\mathcal{R}_k$ and the new function value $\varphi(x^{k+1})$. This method is called the *mean-rule*. We show in the upcoming result Lemma 3.3 that (3.7) holds. Hence, the $\mathcal{R}_k$ chosen by the mean-rule are indeed an upper bound for $\varphi(x^k)$ in our setting. The algorithmic details are presented in Algorithm 3.2.

Algorithm 3.2 (Nonmonotone Projected Subgradient Method).

Require: $x^0 \in C$, $0 < \tau_{\min} \leq \tau_{\max} < \infty$, $\sigma, \beta \in (0, 1)$, $p_{\min} \in (0, 1]$.

```

1: Set  $\mathcal{R}_0 := \varphi(x^0)$ .
2: for  $k = 0, 1, 2, \dots$  do
3:   Choose  $w^k \in \partial\varphi(x^k)$ ;
4:   if suitable termination criterion holds then
5:     STOP and return  $x^k$ .
6:   end if
7:   Choose  $\tau_k \in [\tau_{\min}, \tau_{\max}]$  and  $y^k \in \Pi_D(x^k - \tau_k w^k)$ 
8:   while  $\varphi(y^k) > \mathcal{R}_k + \sigma \langle w^k, y^k - x^k \rangle$  do
9:      $\tau_k = \beta \tau_k$ 
10:    Choose  $y^k \in \Pi_D(x^k - \tau_k w^k)$ .
11:  end while
12:  Set  $x^{k+1} := y^k$ .
13:  Choose  $p_{k+1} \in [p_{\min}, 1]$  and set  $\mathcal{R}_{k+1} := (1 - p_{k+1})\mathcal{R}_k + p_{k+1}\varphi(x^{k+1})$ .
14: end for
```

Let us first collect some properties of the sequence generated by Algorithm 3.2. These are similar to those obtained for a nonmonotone proximal gradient method in [8], where the same linesearch rules were used.

Lemma 3.3. *Let $\varphi : \mathbb{X} \rightarrow \mathbb{R}$ be upper- $\mathcal{C}^2$ on $\mathbb{X}$ with $\inf \varphi > -\infty$. Then for all $x^0 \in \mathbb{X}$, Algorithm 3.2 either stops at some stationary point after a finite number of iterations, or the sequence $\{x^k\}_{k \in \mathbb{N}}$ satisfies the following properties:*

(a) For all $k \in \mathbb{N}$ it holds that

$$\varphi(x^{k+1}) + (1 - p_{k+1})\delta_k \leq \mathcal{R}_{k+1} \leq \mathcal{R}_k - p_{k+1}\delta_k, \quad (3.7)$$

where

$$\delta_k := -\sigma \langle w^k, x^{k+1} - x^k \rangle \geq \frac{\sigma}{2\tau_{\max}} \|x^{k+1} - x^k\|^2 \geq 0, \quad (3.8)$$

where the inequality follows from (2.4).

(b) The sequence $\{\mathcal{R}_k\}$ is monotonically decreasing.

(c) Both $\{\mathcal{R}_k\}$ and $\{\varphi(x^k)\}$ converge to some value $\varphi^* \in \mathbb{R}$, i.e., both sequences converge and have the same limit.

(d) $\|x^{k+1} - x^k\| \rightarrow 0$ for $k \rightarrow \infty$.

Proof. The first inequality of part (a) follows from the definition of $\mathcal{R}_{k+1}$ and the linesearch termination criterion in (3.6):

$$\begin{aligned}\mathcal{R}_{k+1} &= (1 - p_{k+1})\mathcal{R}_k + p_{k+1}\varphi(x^{k+1}) \\ &\geq (1 - p_{k+1})(\varphi(x^{k+1}) - \sigma\langle w^k, x^{k+1} - x^k \rangle) + p_{k+1}\varphi(x^{k+1}) \\ &= \varphi(x^{k+1}) - (1 - p_{k+1})\sigma\langle w^k, x^{k+1} - x^k \rangle.\end{aligned}$$

Similarly, we verify the second inequality:

$$\begin{aligned}\mathcal{R}_{k+1} &= (1 - p_{k+1})\mathcal{R}_k + p_{k+1}\varphi(x^{k+1}) \\ &\leq (1 - p_{k+1})\mathcal{R}_k + p_{k+1}(\mathcal{R}_k + \sigma\langle w^k, x^{k+1} - x^k \rangle) \\ &= \mathcal{R}_k + p_{k+1}\sigma\langle w^k, x^{k+1} - x^k \rangle.\end{aligned}$$

The second assertion directly follows from part (a) and we obtain for all k that

$$\varphi(x^k) \leq \mathcal{R}_k \leq \dots \leq \mathcal{R}_0 = \varphi(x^0).$$

The last equation also shows that, as φ is bounded from below, the sequence of reference values $\{\mathcal{R}_k\}$ is convergent to some limit φ^* . Now, as $p_k \geq p_{\min}$ and

$$\varphi(x^k) = \frac{1}{p_k}(\mathcal{R}_k - (1 - p_k)\mathcal{R}_{k-1}) = \frac{1}{p_k}(\mathcal{R}_k - \mathcal{R}_{k-1}) + \mathcal{R}_{k-1}$$

due to the update of $\mathcal{R}_k$, the convergence of $\{\varphi(x^k)\}$ to the same limit φ^* follows.

Statement (d) is now a consequence of part (a) and the following telescoping argument:

$$\infty > \mathcal{R}_0 - \varphi^* \geq \sum_{j=1}^k (\mathcal{R}_{j-1} - \mathcal{R}_j) \geq \sum_{j=1}^k p_j \delta_{j-1} \geq \sum_{j=1}^k p_{\min} \frac{\sigma}{2\tau_{\max}} \|x^j - x^{j-1}\|^2, \quad (3.9)$$

where we used (3.7) and (3.8). The upper bound $\mathcal{R}_0 - \varphi^*$ does not depend on k and therefore holds for all $k \in \mathbb{N}$. $\square$

We note that (3.9) in the proof above implies the following estimate similar to [1, Proposition 4.4]:

Proposition 3.4. *There exists a constant $c > 0$ such that*

$$\min_{0 \leq j \leq k} \|x^{j+1} - x^j\| \leq \frac{c\sqrt{\mathcal{R}_0 - \varphi^*}}{\sqrt{k+1}}, \text{ for all } k \in \mathbb{N}. \quad (3.10)$$

Proof. By (3.9), we obtain

$$\min_{0 \leq j \leq k} \|x^{j+1} - x^j\|^2 \leq \frac{1}{k+1} \frac{2\tau_{\max}}{p_{\min}\sigma} \sum_{j=1}^{k+1} p_{\min} \frac{\sigma}{2\tau_{\max}} \|x^j - x^{j-1}\|^2 \leq \frac{1}{k+1} \frac{2\tau_{\max}}{p_{\min}\sigma} (\mathcal{R}_0 - \varphi^*).$$

Taking the square root gives the claim with $c := \sqrt{\frac{2\tau_{\max}}{p_{\min}\sigma}}$. $\square$

Next, we establish our global convergence result for the sequence $\{x^k\}$ generated by Algorithm 3.2. This result is inspired by [20], where the smooth case was considered.

Theorem 3.5. *Let $\varphi : \mathbb{X} \rightarrow \mathbb{R}$ be upper- $\mathcal{C}^2$ with $\inf \varphi > -\infty$. Let $\{x^k\}$ be a sequence generated by Algorithm 3.2, given some $x^0 \in \mathbb{X}$. Then, either the method stops at a point with $-w^k \in N_D^P(x^k)$ after a finite number of iterations, or the following assertions hold for $\{x^k\}$:*

1. *Any accumulation point x^* of the sequence of iterates is a proximal critical point. That is, there exists an $w^* \in \partial\varphi(x^*)$ such that $-w^* \in N_D^P(x^*)$.*
2. *Let x^* be an accumulation point of $\{x^k\}$. Then the sequence $\{\varphi(x^k)\}$ converges to $\varphi(x^*)$ and the sequence $\{\mathcal{R}_k\}$ converges monotonically to $\varphi(x^*)$.*

Proof. Let $\{x^k\}_{k \in K}$ be a subsequence converging to $x^* \in C$. Let $\rho \in (0, \infty)$ and define $\bar{\rho}$, τ^* and the interval I as before in (3.5). For all $k \in K$ sufficiently large, it holds that $x^k \in \bar{B}_\rho(x^*)$ and hence, we have $x^{k+1} \in \Pi_D(x^k - \tau^k w^k)$ for $\tau^k \in I$. This gives in particular

$$\|x^{k+1} - (x^k - \tau^k w^k)\| = \text{dist}(x^{k+1} - \tau^k w^k, D). \quad (3.11)$$

By taking a further subsequence, we can assume that τ^k converges to some $\tau > 0$, as I is compact and that $\{w^k\}$ converges as $w^k \rightarrow w^*$ by (local) boundedness of the subdifferential. By Lemma 3.3 it holds that x^{k+1} converges to x^* , too. Now, taking the limit on the subsequence, we obtain from (3.11) that

$$\|x^* - (x^* - \tau w^*)\| = \text{dist}(x^* - \tau w^*, D), \quad (3.12)$$

by continuity of the distance function. This shows that $-w^* \in N_D^P(x^*)$. As w^* is the limit of the subsequence $w^k \in \partial\varphi(x^k)$, possibly on a further subsequence, the robustness property of the limiting (or Clarke) subdifferential gives $w^* \in \partial\varphi(x^*)$. Together with $-w^* \in N_D^P(x^*)$, we get $0 \in \partial\varphi(x^*) + N_D^P(x^*)$.

We already know that $\{\varphi(x^k)\}$ converges. However, on a subsequence, we have $\varphi(x^k) \rightarrow_K \varphi(x^*)$ by continuity of φ . Thus, the last claim concerning the convergence of $\{\varphi(x^k)\}$ and $\{\mathcal{R}_k\}$ follows. $\square$

The proof of Theorem 3.5 reveals that the assertions above follow from two crucial properties: First, locally, the stepsize τ_k stays in some compact interval $I \subseteq \mathbb{R}_+$ and secondly, it holds that $\|x^{k+1} - x^k\| \rightarrow 0$. In the next section, under additional assumptions, we present a linesearch-free method that shares these properties.

4 Auto-Conditioned Projected Subgradient Method

In this section, we also propose an auto-conditioned projected subgradient method and derive corresponding convergence guarantees. The term auto-conditioned refers to the fact that, in particular, our method does not need any problem dependent parameters (such as the constant κ) and does not utilize any linesearch procedures. This type of stepsize strategies use a direct formula to compute the stepsize. Probably the most prominent example in the literature is the adaptive gradient method for convex optimization due to [17]. The method in [17] is fully adaptive and does not assume a global Lipschitz constant for the gradient. However, in the nonconvex case, corresponding results are harder to obtain. Our method is inspired by [16] and [26], where the same stepsize strategy we use was employed in the context where φ is a differentiable and globally Lipschitz-smooth function. The latter assumption seems to be necessary for nonconvex objectives. In the smooth setting, the common motivation behind these auto-conditioned algorithms is that their stepsizes are calculated using a formula to approximate (a multiple of) the inverse of the (local) Lipschitz constant of the gradient. Our method is

presented in Algorithm 4.1. Note further that in contrast to the linesearch we introduced in the preceding section, the auto-conditioned stepsize is decreasing, making the method less adaptive to the local geometry. The main advantage, however, is that no possibly costly backtracking steps are employed. Algorithm 3.2 features a two-loop structure, whereas for Algorithm 4.1 a single loop is sufficient.

Algorithm 4.1 (Auto-Conditioned Projected Subgradient Method).

Require: $x^0 \in C$, $\kappa_0 > 0$, $\alpha > 1$.

1: **for** $k = 0, 1, 2, \dots$ **do**
2: Choose $w^k \in \partial\varphi(x^k)$;
3: **if** suitable termination criterion holds **then**
4: STOP and return x^k .
5: **end if**
6: Set $\gamma_k = \max\{\kappa_0, \dots, \kappa_k\}$ and $\tau_k = \frac{1}{2\alpha\gamma_k}$.
7: Choose $x^{k+1} \in \Pi_D(x^k - \tau_k w^k)$.
8: Set

$$\kappa_{k+1} = \frac{\varphi(x^{k+1}) - \varphi(x^k) - \langle w^k, x^{k+1} - x^k \rangle}{\|x^{k+1} - x^k\|^2}. \quad (4.1)$$

9: **end for**

The formula (4.1) is motivated by part (b) in Proposition 2.2, and provides an estimate for the constant κ . We will assume that Algorithm 4.1 generates an infinite sequence of iterates. In particular, if $\|x^{k+1} - x^k\| = 0$, then by (2.6), x^k is already proximal critical and the algorithm can be stopped before κ_{k+1} is calculated in (4.1).

In the motivating work [26], the Algorithm 4.1 was proposed for a proximal gradient method for composite problems, where one part of the objective is smooth. Let us note that the following property is independent of the choice of descent direction and hence carries over to our setting. First, observe that the sequence $\{\gamma_k\}$ is monotonically increasing and introduce the notation

$$S := \left\{ k \in \mathbb{N} \mid \frac{\alpha + 1}{2} \gamma_k \geq \kappa_{k+1} \right\} \quad \text{and} \quad \bar{S} := \mathbb{N} \setminus S.$$

The following result is now due to [26]. For completeness, the proof is recalled here.

Lemma 4.2. *Let $\{x^k\}$ be the sequence generated by Algorithm 4.1. Then it holds that*

$$\frac{\alpha - 1}{2} \sum_{j=0}^k \gamma_j \|x^{j+1} - x^j\|^2 \leq \varphi(x^0) - \varphi(x^{k+1}) + \sum_{j \in \{0, \dots, k\} \cap \bar{S}} (\gamma_{j+1} - \gamma_j) \|x^{j+1} - x^j\|^2.$$

Proof. Recall that $x^{k+1} \in \Pi_D(x^k - \tau_k w^k)$ is equivalent to

$$x^{k+1} \in \arg \min_{y \in D} \|y - x^k + \tau_k w^k\|^2 = \arg \min_{y \in D} \frac{1}{2\tau_k} \|y - x^k\|^2 + \langle w^k, y - x^k \rangle.$$

Hence, we obtain from $\alpha\gamma_k = \frac{1}{2\tau_k}$ that

$$\alpha\gamma_k \|x^{k+1} - x^k\|^2 + \langle w^k, x^{k+1} - x^k \rangle \leq 0,$$

by choosing $y = x^k$ in the equation above. On the other hand, the definition of κ_{k+1} gives

$$\langle w^k, x^{k+1} - x^k \rangle = \varphi(x^{k+1}) - \varphi(x^k) - \kappa_{k+1} \|x^{k+1} - x^k\|^2.$$

Together, these two equations yield

$$(\alpha\gamma_k - \kappa_{k+1}) \|x^{k+1} - x^k\|^2 + \varphi(x^{k+1}) - \varphi(x^k) \leq 0. \quad (4.2)$$

Now, we consider two cases: First, for $k \in S$, (4.2) yields

$$\begin{aligned}\varphi(x^k) - \varphi(x^{k+1}) &\geq (\alpha\gamma_k - \kappa_{k+1}) \|x^{k+1} - x^k\|^2 \\ &\geq \left(\alpha\gamma_k - \left(\frac{\alpha+1}{2} \gamma_k \right) \right) \|x^{k+1} - x^k\|^2 = \frac{\alpha-1}{2} \gamma_k \|x^{k+1} - x^k\|^2.\end{aligned}\quad (4.3)$$

In the second case, namely $k \in \bar{S}$, we have $\gamma_{k+1} = \max\{\kappa_0, \dots, \kappa_{k+1}\} = \kappa_{k+1}$ as $\alpha > 1$. Hence, applying (4.2) gives

$$\begin{aligned}(\alpha-1)\gamma_k \|x^{k+1} - x^k\|^2 &= (\alpha\gamma_k - \kappa_{k+1} + \gamma_{k+1} - \gamma_k) \|x^{k+1} - x^k\|^2 \\ &\leq \varphi(x^k) - \varphi(x^{k+1}) + (\gamma_{k+1} - \gamma_k) \|x^{k+1} - x^k\|^2.\end{aligned}$$

Combining both cases and taking the sum, the assertion follows. $\square$

However, in order to derive convergence guarantees for our auto-conditioned subgradient method, we require the following stronger assumption for the objective function φ , which guarantees that the stepsizes stay bounded away from zero.

Assumption 4.3. *Assume $C \subseteq \mathbb{X}$ is nonempty and closed and that φ is upper- $\mathcal{C}^2$ such that (2.1) holds globally on C . Further, assume that $\inf \varphi > -\infty$.*

Assumption 4.3 holds in particular if D is a compact set, see Lemma 2.3. Further, under Assumption 4.3, it holds that $\kappa_k \leq \kappa$ for all $k \geq 1$, where κ is now a global constant on D . As claimed, we observe that this implies boundedness of the stepsizes τ_k , namely, we have that for all k that

$$\tau_k \in \left[\frac{1}{2\alpha \max\{\kappa_0, \kappa\}}, \frac{1}{2\alpha \kappa_0} \right].$$

In [26] it was noted that the set $\bar{S}$ is finite. This can be seen as follows: For $k \in \bar{S}$ we have that $\frac{\alpha+1}{2}\gamma_k < \kappa_{k+1} = \gamma_{k+1} \leq \max\{\kappa_0, \kappa\}$. Assuming there is an infinite number of steps, where $k \in \bar{S}$, however, would mean that $\kappa_k \rightarrow \infty$, as $\frac{\alpha+1}{2}$ is a constant greater than 1. This gives a contradiction.

Let us first derive an estimate similar to the corresponding result in Proposition 3.4 for our linesearch method.

Proposition 4.4. *There exists some $c > 0$ depending only on the iterates $x_0, \dots, x^{\max \bar{S}+1}$ and in particular independent of k such that*

$$\min_{0 \leq j \leq k} \|x^{j+1} - x^j\| \leq \frac{c}{\sqrt{k+1}}, \text{ for all } k \in \mathbb{N}. \quad (4.4)$$

Proof. From the previous lemma and as $\kappa_0 \leq \gamma_j$, we have

$$\begin{aligned}\frac{(\alpha-1)(k+1)\kappa_0}{2} \min_{j \in \{0, \dots, k\}} \|x^{j+1} - x^j\|^2 &\leq \frac{\alpha-1}{2} \sum_{j=0}^k \gamma_j \|x^{j+1} - x^j\|^2 \\ &\leq \varphi(x^0) - \varphi(x^{k+1}) + \sum_{j \in \{0, \dots, k\} \cap \bar{S}} (\gamma_{j+1} - \gamma_j) \|x^{j+1} - x^j\|^2,\end{aligned}$$

and $\varphi(x^0) - \varphi(x^{k+1}) \leq \varphi(x^0) - \varphi^*$, where $\varphi^* = \inf \varphi$.

Due to the finiteness of $\bar{S}$, the sum in the last term can be bounded by a finite value independent of k . The claim follows by rearranging terms. $\square$

Further, Algorithm 4.1 has the following properties, which are direct consequences of $\bar{S}$ being finite.

Proposition 4.5. *Under Assumption 4.3, the following assertions hold:*

1. *The sequence $\{\varphi(x^k)\}$ converges to a finite value.*
2. *It holds that $\|x^{k+1} - x^k\| \rightarrow 0$.*

Proof. As $\bar{S}$ is finite, for all k large enough, (4.3) holds. Hence,

$$0 \leq \frac{\alpha - 1}{2} \kappa_0 \|x^{k+1} - x^k\|^2 \leq \varphi(x^k) - \varphi(x^{k+1}).$$

Both claims now follow directly by boundedness of φ from below. $\square$

Finally, this allows us to obtain a global convergence result as before.

Theorem 4.6. *Let Assumption 4.3 hold. Then, given any $x^0 \in \mathbb{X}$, the results from Theorem 3.5 hold for the sequence generated by the method in Algorithm 4.1 (except for the convergence of $\{\mathcal{R}_k\}$, which is not available in our setting).*

Proof. The proof is the same as for Theorem 3.5. $\square$

5 Numerical Results

We now want to evaluate the algorithmic performance of the nonmonotone projected subgradient method from Algorithm 3.2 as well as the auto-conditioned one from Algorithm 4.1. We implemented both algorithms in Python and selected three insightful and relevant examples for testing. In both implementations we used a stopping criterion modeled after the optimality condition (2.6), namely terminating if

$$\|x^k - \Pi_D(x^k - \tau w^k)\|_\infty \leq \varepsilon \quad (5.1)$$

for small $\varepsilon, \tau > 0$, where $w^k \in \partial\varphi(x^k)$. Note that if (5.1) holds with $\varepsilon = 0$, then the iterate x^k is indeed already a proximal critical point.

5.1 MPEC-style model problem

Mathematical programs with equilibrium/complementarity constraints (MPECs or MPCCs) are notoriously difficult to optimize with standard methods, as their feasible set may contain points that do not fulfill any common constraint qualification [22]. For this problem class, one can construct examples, for which the different normal cones introduced in Section 2 do not coincide, and thus algorithms which are searching for points with weaker stationarity may converge to suboptimal solutions. In the following example, we can identify three points with different stationarity properties, hence we can illustrate the advantageous theoretical guarantees of our projected subgradient method in this case.

To this end, we consider the following two-dimensional optimization problem

$$\min_{(x_1, x_2) \in \mathbb{R}^2} f(x_1, x_2) := \frac{1}{2}(x_1 - 1)^2 + \frac{1}{2}(x_2 - 1)^2 \quad \text{s.t.} \quad x \in D, \quad (5.2)$$

where the feasible set D is defined as

$$D := ([0, \infty) \times \{0\}) \cup (\{0\} \times [0, 2]) \cup \{(t, t+2) \mid t \geq 0\},$$

cf. Figure 1. This example is a modification of the MPEC used in [22, Example 10]. The feasible set is obviously non-empty and closed, also by the construction as the union of three closed, convex, non-empty sets, the projection of x onto D can be easily calculated by first projecting

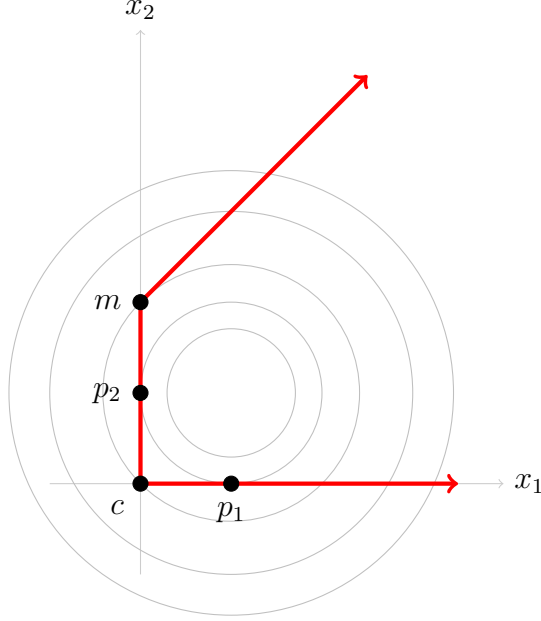


Figure 1: Feasible set of the MPEC-style problem with contours of the objective.

onto each of the three line segments, and then just comparing the distances. The projections onto the line segments with minimal distance are then the set members of $\Pi_D(x)$.

The objective function f describes a two-dimensional paraboloid centered around $(1, 1)$. In particular, the objective function is smooth, hence the notions of criticality as defined in Section 2 and stationarity, as defined for example in [20], coincide. Thus, in the following we use the more commonly used term $*$ -stationarity to describe the optimality conditions

$$0 \in \nabla f(\bar{x}) + N_D^*(\bar{x})$$

for $*$ $\in \{P, F, M, C\}$.

By the monotonicity of the objective, we see that f attains its minimal value on D in the closest points to $(1, 1)$, which are $p_1 := (1, 0)$ and $p_2 := (0, 1)$. By Proposition 2.5, both points must be P-stationary, and hence also F- and M-stationary. The point $c := (0, 0)$ is non-optimal, but C-stationary, which holds analogously to [22, Example 10]. In fact, Scheel and Scholtes showed the stricter property that c is even C-stationary in the MPEC sense. We now show that the point $m := (0, 2)$ is M-stationary, but not F-stationary, hence also suboptimal.

We have

$$\nabla f(m) = \begin{pmatrix} -1 \\ 1 \end{pmatrix},$$

hence we want to show that $(1, -1)$ lies in $N_D^M(m)$. We define the sequence $x^k := (1/k, 2 + 1/k) \rightarrow m$. In order to determine $N_D^F(x^k)$, one can see that, as each x^k lies on the line segment $\{(t, t+2) \mid t > 0\}$, the tangent cone consists only of the direction of the line segment, namely $T_D(x^k) = \{(t, t) \mid t \in \mathbb{R}\}$. We have for the polar cone

$$N_D^F(x^k) = T_D(x^k)^\circ = \{(t, -t) \mid t \in \mathbb{R}\}.$$

In particular, this shows $(1, -1) \in N_D^F(x^k)$ for each $k \in \mathbb{N}$. By defining $v^k \equiv (1, -1)$, this shows

$$-\nabla f(m) \in N_D^M(m)$$

and thus the M-stationarity of m . However, we have $d := (0, -1) \in T_D(m)$, and thus

$$-\nabla f(m)^T d = 2 > 0,$$

Method	LS	AC
converging to p_1	4687	4864
converging to p_2	5313	5136

Table 1: Convergence of starting grid values for the nonmonotone linesearch (LS) and the auto-conditioned stepsize (AC)

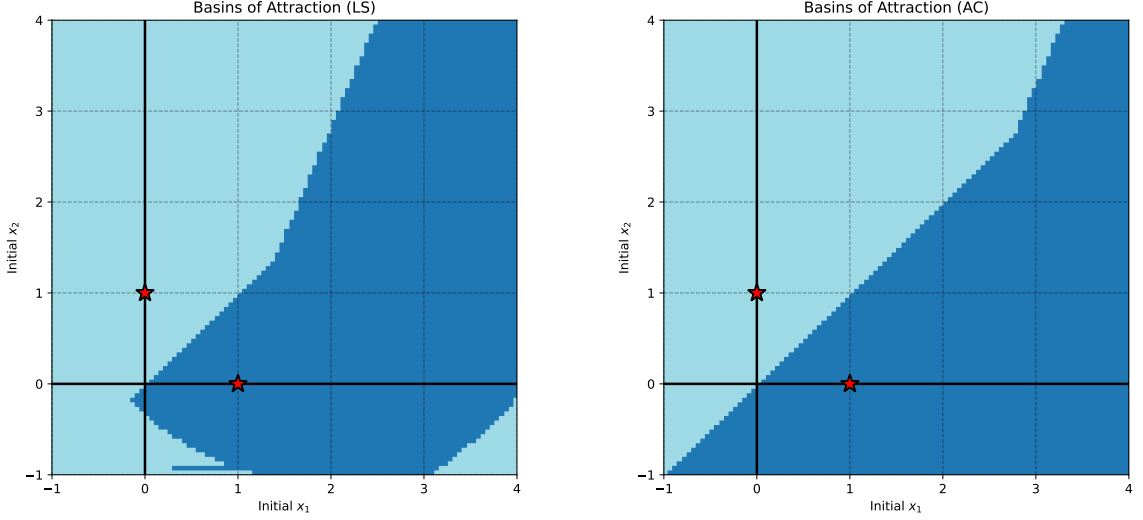


Figure 2: Basins of attraction for the grid of starting values. Dark area: convergence to p_1 , light area: convergence to p_2

which shows that m is not F-stationary.

This problem is an example for a setting, where using a method searching only for M- or C-stationary points may lead to suboptimal results, and the theoretical guarantees provided in Sections 3 and 4 matter. For numerical confirmation, we ran both projected gradient methods on a grid of 10000 starting values equidistantly spaced in the box $[-1, 4]^2$. We set $\varepsilon = 10^{-6}$ and $\tau = 0.1$. In all instances, the projected gradient method converges to one of the optimal P-stationary points p_1 or p_2 , as displayed in Table 1 and visualized in Figure 2.

5.2 MAXCUT

The MAXCUT problem is an NP-hard [11] graph theoretical problem, we follow the definition from [14]: Let $G := (V, E)$ be a weighted undirected graph with vertex set $V = \{1, \dots, n\}$ and edges e_{ij} connecting $i, j \in V$. Let $A = (a_{ij})$ be the (symmetric) adjacency matrix, where an entry $a_{ij} = a_{ji}$ denotes the non-negative weight of e_{ij} , which can also be set to zero. We now define a *cut* for $S \subseteq V$ as the set

$$\delta(S) := \{e_{ij} \mid i \in S, j \in \bar{S}\},$$

with $\bar{S} = V \setminus S$. This describes the set of all edges which connect a point in S to a point in $\bar{S}$. The weight of the cut is defined by

$$w(S) := \sum_{e_{ij} \in \delta(S)} a_{ij}.$$

Our goal is now to find the cut with maximum weight. Following the reasoning of [11], the problem is transformed to a semidefinite program by assigning a value $w_i \in \{-1, +1\}$ for each

vertex $v_1, \dots, v_n$, where $w_i = 1$ if $v_i \in S$ and $w_i = -1$ if $v_i \in \bar{S}$. Now, instead of having w_i with $|w_i| = 1$ be elements of the unit sphere in one dimension, the problem is generalized by considering vectors $y_i \in \mathbb{R}^n$ with $\|y_i\|_2 = 1$. Defining the matrix $Y \in \mathbb{R}^{n \times n}$ as (y_{ij}) with $y_{ij} = y_i^T y_j$, we obtain the semidefinite relaxation to the MAXCUT problem via $L := \text{diag}(Ae) - A$ by

$$\min_{Y \in \mathbb{R}_{\text{sym}}^{n \times n}} -\frac{1}{4} \text{trace}(LY) \quad \text{s.t.} \quad \text{diag } Y = e, Y \succeq 0. \quad (5.3)$$

The solution of this problem is matrix-valued and has to be transformed back in order to obtain a cut vector, for details see [11]. However, if we add the additional constraint $\text{rank } Y = 1$ to problem (5.3), a feasible Matrix Y is a symmetric matrix of rank one with $\text{diag } Y = e$, which results in $Y = ww^T$ with $w \in \{-1, 1\}^n$. Hence by adding the rank constraint and determining the vector w , we recuperate a solution of the original MAXCUT problem. Our resulting optimization problem is now

$$\min_{W \in \mathbb{R}_{\text{sym}}^{n \times n}} -\frac{1}{4} \text{trace}(LW) \quad \text{s.t.} \quad \text{diag } W = e, W \succeq 0, \text{rank } W = 1. \quad (5.4)$$

As the feasible set is still not tractable for projection, we modify the problem to

$$\min_{W \in \mathbb{R}_{\text{sym}}^{n \times n}} -\frac{1}{4} \text{trace}(LW) + \rho \left(\text{trace}(W) - \|W\|_2 \right) \quad \text{s.t.} \quad \text{diag } W = e, W \succeq 0 \quad (5.5)$$

with a penalty parameter $\rho > 0$. The penalty term is constructed utilizing several properties of the problem. Let $\lambda_1(W) \geq \lambda_2(W) \geq \dots \geq \lambda_n(W) \geq 0$ be the eigenvalues of the symmetric positive semidefinite matrix W sorted by value from largest to smallest.

- By symmetry and positive semidefiniteness we have $\|W\|_2 = \lambda_1(W)$ and

$$\text{trace}(W) = \sum_{i=1}^n \lambda_i(W) \geq 0$$

Hence we obtain the expression

$$\text{trace}(W) - \|W\|_2 = \sum_{i=2}^n \lambda_i(W)$$

and we penalize the $n - 1$ smallest eigenvalues of W in order to obtain a matrix of rank ≤ 1 .

- By the diagonal constraint $\text{diag } W = e$, the zero-matrix is not feasible, and the penalty term is a sensible choice for enforcing matrix rank 1.

Next we want to certify that the problem formulation (5.5) fits into our projected gradient framework. First, owing to Proposition 2.2 (c), we see that our objective function is upper- $\mathcal{C}^2$ by writing

$$g(W) := \text{trace} \left(\left(-\frac{1}{4}L + \rho I \right) W \right), \quad h(W) = \rho \|W\|_2$$

and noting that g as a linear function is differentiable with Lipschitz gradient and h as a scaled norm is convex. Furthermore, by setting the objective $f = g - h$, the limiting subgradient is given by

$$\partial f(W) = -\frac{1}{4}L + \rho I - \rho \left\{ vv^T \mid v \in E_{\lambda_1(W)}, \|v\| = 1 \right\},$$

where $E_{\lambda_1(W)}$ denotes the eigenspace of the largest eigenvalue of W .

Projecting onto the feasible set

$$D := \left\{ W \in \mathbb{R}^{n \times n} \mid W \text{ symmetric}, W \succeq 0, \text{diag } W = e \right\}$$

has no explicit form, but it is a common problem known as *Nearest Correlation Matrix*, and there exist numerically efficient ways to compute this projection. In our implementation, we used a method by Borsdorf and Higham, which in turn is a preconditioned form of a semismooth Newton method, cf. [5].

As a test problem collection, we used the **rudu** dataset from the MAXCUT problems in the Biq Mac library¹, which consists of 130 randomly generated graphs. The optimal values of the problems are known. In order to obtain a good starting value for our projected subgradient method, we ran the SCS solver for semidefinite programs from the **cvxpy** package [9] on the semidefinite convex relaxation (5.3), and fed the output into our algorithms. We ran all problem instances with a tolerance of $\varepsilon = 10^{-6}$ and chose $\tau = 0.1$ as well as $\rho = 5$.

Algorithms 3.2 and 4.1 terminated successfully on all problem instances. The solutions of the two algorithms coincide on all problems and achieve a lower relative error than the solution of the semidefinite relaxation. A more detailed evaluation of the accuracies of our PG implementation versus the semidefinite relaxation is given in Figure 3. Here we can also see the superior

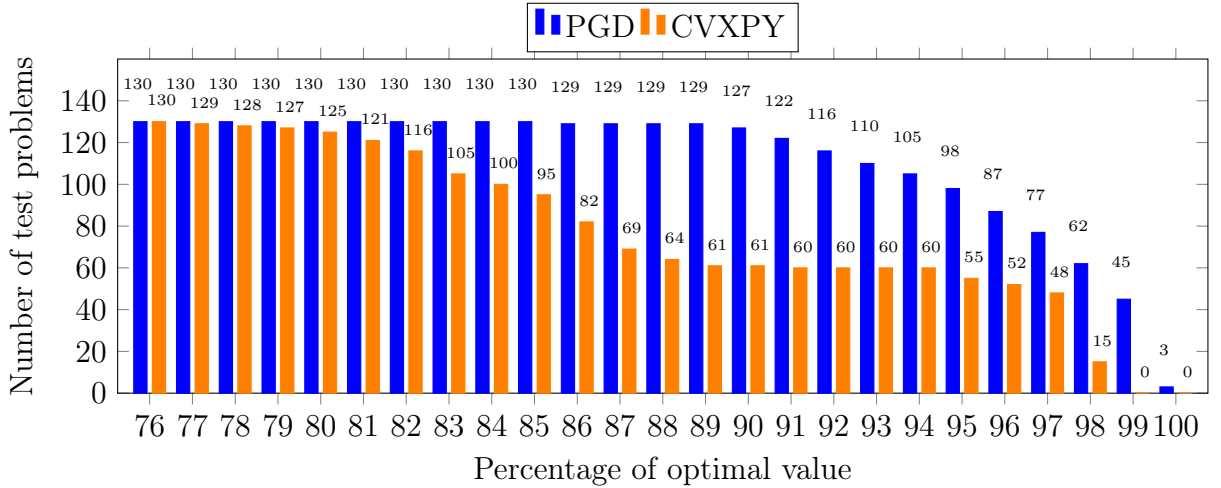


Figure 3: Accuracies for the MAXCUT problems from the **rudu** dataset in the Biq Mac library.

performance of PGD on the **rudu** collection. For example, our method is capable of solving 62 of the 130 problems with an accuracy of at least 98%, where the semidefinite relaxation achieves this accuracy only on 15 problems. We also note that, as in the reasoning above, the results of the semidefinite relaxation cannot be directly interpreted as a solution to MAXCUT, we observed them to be even infeasible to problem (5.4) as they generally have full rank.

Comparing both PG approaches, we display the metrics of CPU time, iterations of the projected subgradient method and of the Newton iterations needed for calculating the projections in Figure 4.

We note that the linesearch performs slightly better than the auto-conditioned variant across all metrics. This may be explained by the fact that we observed quick acceptance of the descent direction in most of the iterations of PG with linesearch, hence the auto-conditioned version cannot save significant computational effort compared to the linesearch in this case.

¹<https://biqmac.aau.at/biqmaclib.html>

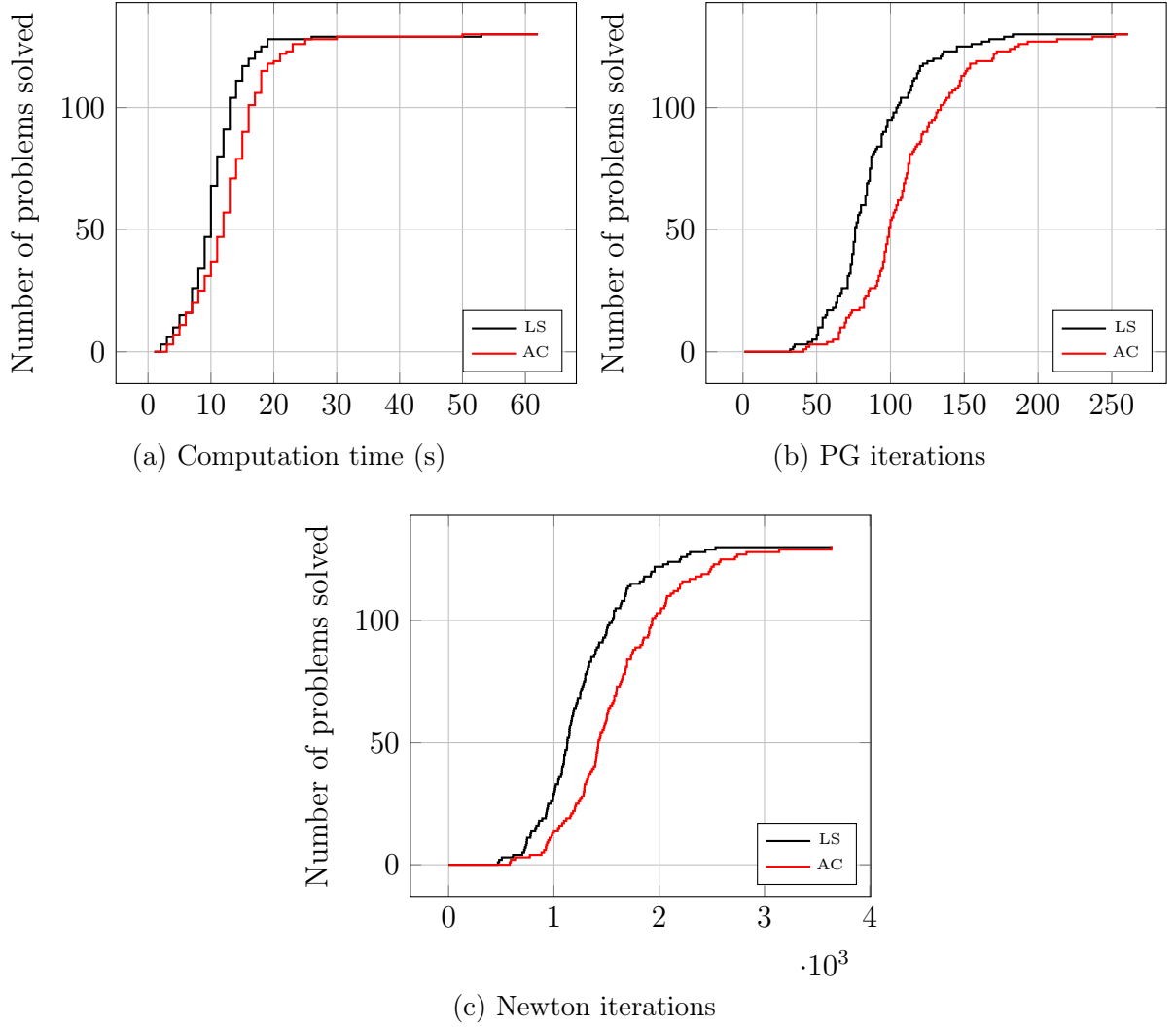


Figure 4: Performance profiles comparing the projected subgradient method with line-search and auto-conditioning on the MAXCUT problems.

5.3 Robust Principal Component Analysis

Robust Principal Component Analysis (Robust PCA) was first introduced in [25] as the following task: Let $Y = L + S$ be a given matrix $Y \in \mathbb{R}^{m \times n}$ that is the sum of an unknown low-rank signal $L \in \mathbb{R}^{m \times n}$ and a sparse error term $S \in \mathbb{R}^{m \times n}$. We can only observe Y and want to reconstruct L and S . This leads to the following optimization problem

$$\min_{L, S \in \mathbb{R}^{m \times n}} \|Y - L - S\|_F \quad \text{s.t.} \quad \text{rank } L \leq r, \|S\|_0 \leq k \quad (5.6)$$

for $0 \leq r \leq \min\{m, n\}$ and $0 \leq k \leq mn$, where $\|S\|_0$ denotes the number of non-zero entries in S . The initial approach in [25] was to convexify the problem, while more recent approaches like the one in [13] tackle problem (5.6) directly. We also want to follow the direct approach, but first have to bring problem (5.6) into a suitable form for our algorithmic framework. First we define the sets

$$D := \{A \in \mathbb{R}^{m \times n} \mid \text{rank } A \leq r\}$$

and

$$M := \{Y - S \mid \|S\|_0 \leq k\}.$$

Then we can reformulate problem (5.6) as

$$\min_{L \in \mathbb{R}^{m \times n}} \text{dist}_M(L)^2 \quad \text{s.t.} \quad L \in D. \quad (5.7)$$

This representation has the following properties: The squared distance function is upper- $\mathcal{C}^2$, as for arbitrary $X \in \mathbb{R}^{m \times n}$, it can be written as

$$\begin{aligned} \text{dist}_M(X)^2 &= \inf_{Y \in M} \|X - Y\|^2 \\ &= \inf_{Y \in M} \left\{ \|X\|^2 - 2\langle X, Y \rangle + \|Y\|^2 \right\} \\ &= \|X\|^2 - \sup_{Y \in M} \left\{ 2\langle X, Y \rangle - \|Y\|^2 \right\}. \end{aligned}$$

Setting

$$g(X) := \|X\|^2, \quad h(X) := \sup_{Y \in M} \left\{ 2\langle X, Y \rangle - \|Y\|^2 \right\},$$

we note that $X \mapsto 2\langle X, Y \rangle - \|Y\|^2$ as an affine function is in particular convex for each $Y \in M$, and thus taking the supremum over all $Y \in M$ implies the convexity of h , cf. [3, Proposition 8.16]. Thus we can write $\text{dist}_M^2 = g - h$, with g differentiable with Lipschitz gradient and h convex, and Proposition 2.2 (c) is again applicable.

The limiting subgradient of the squared distance function is well known, we have

$$\partial \text{dist}_M(X)^2 = X - \Pi_M(X)$$

cf. [10, Lemma 7]. Projecting a matrix $X \in \mathbb{R}^{m \times n}$ onto the matrices with rank less than r is also a standard task and is done by truncating the singular value decomposition of X to the largest r singular values, and setting the rest to zero, cf. [19, Section 6.2]. In order to calculate the projection onto M , which is used in the calculation of dist_M^2 as well as its subgradient, we note that by construction an element $W \in \Pi_M(X)$ is obtained by taking the k largest absolute value entries in $Y - X$ and setting the others to zero.

Name	PGD iterations	CPU time in seconds
PETS2006 LS	16	109.56
PETS2006 AC	17	116.47
highway LS	3	20.23
highway AC	3	19.87
skating LS	9	62.14
skating AC	8	56.32

Table 2: Algorithmic Data for the robust PCA example

In order to test the robust PCA implementation, we used the dataset **CDnet2014**²[24], which consists of different image sequences from videos, mostly from static surveillance camera footage. On this data, we can use robust PCA to do video background subtraction. We construct our observation matrix $Y \in \mathbb{R}^{m \times n}$ by setting its columns to a vector of length $m = pq$ containing the flattened version of each gray scale video frame of resolution $p \times q$. The number n of columns in Y is then the number of frames in the sequence we want to analyze. The intuition for using robust PCA to separate the video background from foreground is that in surveillance camera footage, the camera is static, hence the background components should have a constant pixel value across the whole observation sequence. Thus, if we only had the background in Y , we would have constant rows, and Y would be a rank-1-matrix. However, the background is perturbed by the dynamic movements in the foreground, which can now be modeled as the sparse error term $S \in \mathbb{R}^{m \times n}$, as we also assume the foreground motion not to completely dominate the image. Hence, in this setting, the task of robust PCA is to determine the low rank background L , which can be used to separate background and foreground in our observation Y .

We ran the projected subgradient method in linesearch and auto-conditioned version on three different image sequences **PETS2006** (288×320), **highway** (240×320) and **skating** (240×360) of length $n = 400$. Starting with a random matrix of the same dimensions as Y , all algorithms converged with desirable visual results displayed in Figure 5 for single example frames. Here we set $r = 2$, $k = mn \cdot 10^{-4}$, $\tau = 10^{-5}$ and $\varepsilon = 10^{-4}$. Computational data is displayed in Table 2. The computation times are overall comparable to the ones in [6] obtained by an accelerated alternating projection algorithm, but should be also read with caution, as they are highly dependent on the problem dimension, which in return scales with the underlying image resolution and sequence length.

6 Final Remarks

In this paper, we introduced the projected subgradient method for a class of nonsmooth objectives, namely upper- $\mathcal{C}^2$ functions. We showed that both presented algorithm versions, the one with a nonmonotone linesearch in Algorithm 3.2 and the auto-conditioned Algorithm 4.1, possess strong convergence guarantees to a proximal critical point under mild assumptions. Furthermore, we validated the theoretical guarantees in a numerical setting and applied the method to solve real world numerical tasks with promising results. Possible directions for our future research include the design of new methods for the considered optimization problem with convergence to points which satisfy even stronger stationarity notions. Furthermore, the investigation of a generalization to proximal (sub-)gradient methods deserves attention. However, as already noted in [20], this would need significantly new techniques.

Funding. No funding was received to assist with the preparation of this manuscript.

²<https://changedetection.net/dataset2014/>

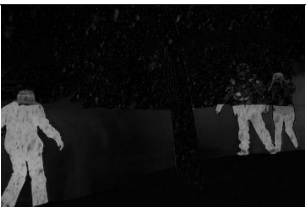
Name	V.	Original image Y	Low-rank background L	$ Y - L $ (pointwise)
PETS2006	LS			
	AC			
highway	LS			
	AC			
skating	LS			
	AC			

Figure 5: RPCA results for video background subtraction.

Code and Data Availability. The source code for the implementation of the projected subgradient methods as well as the MPEC style, MAXCUT and Robust PCA numerical examples is available at <https://github.com/jakrueg/projected-subgradient>, the problem data can be sourced from <https://biqmac.aau.at/biqmaclib.html> and <https://changedetection.net/dataset2014/> respectively.

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

References

- [1] Francisco J. Aragón-Artacho, Rubén Campoy, Pedro Pérez-Aros, and David Torregrosa-Belén. *Nonmonotone subgradient methods based on a local descent lemma*. 2025. arXiv: 2510.19341 [math.OC].
- [2] Larry Armijo. “Minimization of functions having Lipschitz-continuous first partial derivatives”. In: *Pacific Journal of Mathematics* 16 (1966), pp. 1–3. DOI: 10.2140/pjm.1966.16.1.
- [3] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. 2nd edition. Cham: Springer, 2017. DOI: 10.1007/978-3-319-48311-5.
- [4] Ernesto G. Birgin, José Mario Martínez, and Marcos Raydan. “Nonmonotone spectral projected gradient methods on convex sets”. In: *SIAM Journal on Optimization* 10.4 (2000), pp. 1196–1211. DOI: 10.1137/S1052623497330963.
- [5] Rüdiger Borsdorf and Nicholas J. Higham. “A preconditioned Newton algorithm for the nearest correlation matrix”. In: *IMA Journal of Numerical Analysis* 30.1 (2010), pp. 94–107. DOI: 10.1093/imanum/drn085.
- [6] HanQin Cai, Jian-Feng Cai, and Ke Wei. “Accelerated Alternating Projections for Robust Principal Component Analysis”. In: *Journal of Machine Learning Research* 20 (2019), 20:1–20:33. URL: <https://jmlr.org/papers/v20/18-022.html>.
- [7] Francis H. Clarke, Yuri S. Ledyaev, Ronald J. Stern, and Peter R. Wolenski. *Nonsmooth Analysis and Control Theory*. Springer New York, NY, 1998. URL: <https://doi.org/10.1007/b97650>.
- [8] Alberto De Marchi. “Proximal gradient methods beyond monotony”. In: *Journal of Nonsmooth Analysis and Optimization* 4 (2023), p. 18. DOI: 10.46298/jnsao-2023-10290.
- [9] Steven Diamond and Stephen Boyd. “CVXPY: a Python-embedded modeling language for convex optimization”. In: *Journal of Machine Learning Research* 17 (2016), pp. 1–5. URL: jmlr.csail.mit.edu/papers/v17/15-408.html.
- [10] Guillaume Garrigos. *Square distance functions are Polyak-Łojasiewicz and vice-versa*. 2023. arXiv: 2301.10332 [math.OC].
- [11] Michel X. Goemans and David P. Williamson. “Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming”. In: *Journal of the Association for Computing Machinery* 42.6 (1995), pp. 1115–1145. DOI: 10.1145/227683.227684.

- [12] Luigi Grippo, Francesco Lampariello, and Stefano Lucidi. “A nonmonotone line search technique for Newton’s method”. In: *SIAM Journal on Numerical Analysis* 23.4 (1986), pp. 707–716. DOI: 10.1137/0723046.
- [13] Keaton Hamm, Mohamed Meskini, and HanQin Cai. “Riemannian CUR decompositions for robust principal component analysis”. In: *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*. Vol. 196. Proceedings of Machine Learning Research. PMLR, 2022, pp. 152–160. URL: <https://proceedings.mlr.press/v196/hamm22a.html>.
- [14] Xiaoxi Jia, Christian Kanzow, Patrick Mehrlitz, and Gerd Wachsmuth. “An augmented Lagrangian method for optimization problems with structured geometric constraints”. In: *Mathematical Programming. Series A. Series B* 199.1-2 (A) (2023), pp. 1365–1415. DOI: 10.1007/s10107-022-01870-z.
- [15] Christian Kanzow and Leo Lehmann. *A nonmonotone descent method for optimization problems defined by upper- C^2 functions over submanifolds*. 2026. arXiv: 2605.26909 [math.OC].
- [16] Guanghui Lan, Tianjiao Li, and Yangyang Xu. *Projected gradient methods for non-convex and stochastic smooth optimization: new complexities and auto-conditioned stepsizes*. 2026. arXiv: 2412.14291 [math.OC].
- [17] Yura Malitsky and Konstantin Mishchenko. “Adaptive gradient descent without descent”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 6702–6712. URL: <http://proceedings.mlr.press/v119/malitsky20a.html>.
- [18] Boris. S. Mordukhovich. *Variational Analysis and Applications*. Springer, 2018. DOI: 10.1007/978-3-319-92775-6.
- [19] Guillaume Olikier, Kyle A. Gallivan, and P. A. Absil. *First-order optimization on stratified sets*. 2023. arXiv: 2303.16040 [math.OC].
- [20] Guillaume Olikier and Irène Waldspurger. “Projected gradient descent accumulates at Bouligand stationary points”. In: *SIAM Journal on Optimization* 35.2 (2025), pp. 1004–1029. DOI: 10.1137/24M1692782.
- [21] R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*. Springer, 2009. DOI: 10.1007/978-3-642-02431-3.
- [22] Holger Scheel and Stefan Scholtes. “Mathematical programs with complementarity constraints: stationarity, optimality, and sensitivity.” In: *Mathematics of Operations Research* 25.1 (2000), pp. 1–22. DOI: 10.1287/moor.25.1.1.15213.
- [23] Hoai An Le Thi, Van Ngai Huynh, and Tao Pham Dinh. *Finding directional stationary points of DC programs*. 2026. arXiv: 2605.15838 [math.OC].
- [24] Yi Wang, Pierre-Marc Jodoin, Fatih Murat Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. “CDnet 2014: An expanded change detection benchmark dataset”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2014, pp. 393–400. DOI: 10.1109/CVPRW.2014.126.

- [25] John Wright, Arvind Ganesh, Shankar R. Rao, YiGang Peng, and Yi Ma. “Robust Principal Component Analysis: Exact recovery of corrupted low-rank matrices via convex optimization”. In: *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., 2009, pp. 2080–2088. URL: <https://proceedings.neurips.cc/paper/2009/hash/c45147dee729311ef5b5c3003946c48f-Abstract.html>.
- [26] Shotaro Yagishita and Masaru Ito. *Simple linesearch-free first-order methods for nonconvex optimization*. 2025. arXiv: 2509.14670 [math.OC].
- [27] Hongchao Zhang and William W. Hager. “A nonmonotone line search technique and its application to unconstrained optimization”. In: *SIAM Journal on Optimization* 14.4 (2004), pp. 1043–1056. DOI: 10.1137/S1052623403428208.