Exterior-point Optimization for Nonconvex Learning

Shuvomoy Das Gupta [*1], Bartolomeo Stellato [†2], and Bart P.G. Van Parys [‡3]

[1]Operations Research Center, Massachusetts Institute of Technology
[2]Department of Operations Research and Financial Engineering, Princeton University
[3]Sloan School of Management, Massachusetts Institute of Technology

December 23, 2021

Abstract

In this paper we present the *nonconvex exterior-point optimization solver* (NExOS)—a novel first-order algorithm tailored to constrained nonconvex learning problems. We consider the problem of minimizing a convex function over nonconvex constraints, where the projection onto the constraint set is single-valued around local minima. A wide range of nonconvex learning problems have this structure including (but not limited to) sparse and low-rank optimization problems. By exploiting the underlying geometry of the constraint set, NExOS finds a locally optimal point by solving a sequence of penalized problems with strictly decreasing penalty parameters. NExOS solves each penalized problem by applying a first-order algorithm, which converges linearly to a local minimum of the corresponding penalized formulation under mild technical conditions. Furthermore, the local minima of the penalized problems converge to a local minimum of the original problem as the penalty parameter goes to zero. We implement NExOS in the open-source `Julia` package `NExOS.jl`, which has been extensively tested on many instances from a wide variety of learning problems. We demonstrate that our algorithm, in spite of being general purpose, outperforms specialized methods on several examples of well-known nonconvex learning problems involving sparse and low-rank optimization. For sparse regression problems, NExOS finds locally optimal solutions which dominate `glmnet` in terms of support recovery, yet its training loss is smaller by an order of magnitude. For low-rank optimization with real-world data, NExOS recovers solutions with 3 fold training loss reduction, but with a *proportion of explained variance* that is 2 times better compared to the nuclear norm heuristic.

1 Introduction

This paper studies machine learning problems involving a convex cost function f in the presence of a quadratic regularizer term $(\beta/2)\|\cdot\|^2$ over a closed nonconvex constraint set $\mathcal{X}$. We propose a novel first-order algorithm *nonconvex exterior-point optimization solver* (NExOS) to solve such problems numerically. We can write such problems compactly as:

$$\text{minimize} \quad f(x) + (\beta/2)\|x\|^2 + \iota(x), \tag{$\mathcal{P}$}$$

where the decision variable x takes value in a finite-dimensional vector space $\mathbf{E}$ over the reals and and $\iota(x)$ denotes the indicator function of the set $\mathcal{X}$ at x, which is 0 if $x \in \mathcal{X}$ and ∞ else.

*Corresponding Author: sdgupta@mit.edu
†bstellato@princeton.edu
‡vanparys@mit.edu

Furthermore, $\mathbf{E}$ is equipped with inner product $\langle \cdot \mid \cdot \rangle$ and norm $\|\cdot\| = \sqrt{\langle x \mid x \rangle}$. The regularization parameter $\beta > 0$ is commonly introduced to reduce the generalization error without increasing the training error [40, §5.2.2]. The constraint set $\mathcal{X}$ is closed, potentially nonconvex, but prox-regular at local minima, *i.e.*, it has single-valued Euclidean projection around local minima [19].

Definition 1 (Prox-regular set [19])**.** A nonempty closed set $\mathcal{S} \subseteq \mathbf{E}$ is prox-regular at a point $x \in \mathcal{S}$ if projection onto $\mathcal{S}$ is single-valued on a neighborhood of x.

1.1 Applications

Among different prox-regular sets, sparse and low-rank constraint sets are perhaps the most prominent in the machine learning literature because they allow for greater interpretability, speed-ups in computation, and reduced memory requirements [18]. Both sets are at the core of many modern methods dealing with high dimensional data.

Low-rank optimization problems are in the form ($\mathcal{P}$); they are critically used in many machine learning problems such as collaborative filtering [18, pp. 279-281], design of online recommendation systems [32, 33], bandit optimization [60], data compression [34, 35, 36], and low rank kernel learning [37]. In these applications, the constraint set $\mathcal{X}$ decomposes as $\mathcal{X} = \mathcal{C} \bigcap \mathcal{N}$, where $\mathcal{C}$ is a compact convex set, and

$$\mathcal{N} = \{X \in \mathbf{R}^{m \times d} \mid \mathbf{rank}(X) \leq k\}, \tag{1}$$

where $\mathcal{N}$ in (1) is prox-regular at any point $X \in \mathbf{R}^{m \times d}$ where $\mathbf{rank}(X) = k$ [25, Proposition 3.8]. One can show that $\mathcal{X}$ inherits the prox-regularity property around local minima from the set $\mathcal{N}$; we provide a formal proof of this statement in Lemma 3 in Appendix A.1. In this paper, we apply NExOS to solve the affine rank minimization problem that can be formulated as:

$$\begin{array}{ll} \text{minimize} & \|\mathcal{A}(X) - b\|_2^2 + (\beta/2)\|X\|_F^2 \\ \text{subject to} & \mathbf{rank}(X) \leq r \\ & \|X\|_2 \leq \Gamma, \end{array} \tag{RM}$$

where $X \in \mathbf{R}^{m \times d}$ is the decision variable, $b \in \mathbf{R}^k$ is a noisy measurement data, and $\mathcal{A} : \mathbf{R}^{m \times d} \to \mathbf{R}^k$ is a linear map. The parameter $\Gamma > 0$ denotes the upper bound for the spectral norm of X. The affine map $\mathcal{A}$ can be determined by k matrices $A_1, \ldots, A_k$ in $\mathbf{R}^{m \times d}$ such that $\mathcal{A}(X) = (\mathbf{tr}(A_1^T X), \ldots, \mathbf{tr}(A_k^T X))$. We will present several numerical experiments to solve (RM) using NExOS for both for both synthetic and real-world datasets in §3.2.

Cardinality or sparsity constraints have found applications in many practical settings, *e.g.*, gene expression analysis [27, pp. 2–4], sparse regression [18, pp. 155–157], signal transmission and recovery [29, 28], hierarchical sparse polynomial regression [30], and best subset selection [31], just to name a few. In these problems, the constraint set $\mathcal{X}$ decomposes as $\mathcal{X} = \mathcal{C} \bigcap \mathcal{N}$, where $\mathcal{C}$ is a compact convex set, and

$$\mathcal{N} = \{x \in \mathbf{R}^d \mid \mathbf{card}(x) \leq k\}, \tag{2}$$

where $\mathbf{card}(x)$ counts the number of nonzero elements in x. Similarly, $\mathcal{N}$ in (2) is prox-regular at any point x satisfying $\mathbf{card}(x) = k$ because we can write $\mathbf{card}(x) \leq k$ as a special case of the low-rank constraint by embedding the components of x in the diagonal entries of a matrix and then using the prox-regularity of low-rank constraint set. In this paper, we apply NExOS to solve the

sparse regression problem for both synthetic and real-world datasets in §3.1, which is concerned with approximating a vector $b \in \mathbf{R}^m$ with a linear combination of at most k columns of a matrix $A \in \mathbf{R}^{m\times d}$ with bounded coefficients. This problem has the form:

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2^2 + (\beta/2)\|x\|_2^2 \\ \text{subject to} & \mathbf{card}(x) \leq k \\ & \|x\|_\infty \leq \Gamma, \end{array} \tag{SR}$$

where $x \in \mathbf{R}^d$ is the decision variable, and $A \in \mathbf{R}^{m\times d}$, $b \in \mathbf{R}^m$, and $\Gamma > 0$ are problem data.

Some other notable prox-regular sets are as follows. Closed convex sets are prox-regular everywhere [12, page 612]. Examples of well-known prox-regular sets that are not convex include weakly convex sets [23], proximally smooth sets [22], strongly amenable sets [12, page 612], and sets with Shapiro property [24]. Also, a nonconvex set defined by a system of finitely many inequality and equality constraints for which a basic constraint qualification holds is prox-regular [50, page 10].

1.2 Related work

Due to the presence of the nonconvex set $\mathcal{X}$, the nonconvex problem ($\mathcal{P}$) is $\mathcal{NP}$-hard [38]. A common way to deal with this issue is to avoid this inherent nonconvexity altogether by convexifying the original problem. The relaxation of the sparsity constraint leads to the popular Lasso formulation and its many variants [27], whereas relaxation of the low-rank constraints produces the nuclear norm based convex models [39]. The basic advantage of the convex relaxation technique is that, in general, a globally optimal solution to a convex problem can be computed reliably and efficiently [15, §1.1], whereas for nonconvex problems a local optimal solution is often the best one can hope for. Furthermore, if certain statistical assumptions on the data generating process hold , then it is possible to recover exact solutions to the original nonconvex problems with high probability by solving the convex relaxations (see [27] and the references therein). However, if these stringent assumptions do not hold, then solutions to the convex formulations can be of poor quality and may not scale very well [18, §6.3 and §7.8]. In this situation, the nonconvexity of the original problem must be confronted directly, because such nonconvex formulations capture the underlying problem structures more accurately than their convex counterparts.

To that goal, first-order algorithms such as hard thresholding algorithms such as IHT, NIHT, HTP, CGIHT, address nonconvexity in sparse and low-rank optimization by implementing variants of projected gradient descent with projection taken onto the sparse and/or low-rank set [2, 3, 4]. While these algorithms have been successful in recovering low-rank and sparse solutions in underdetermined linear systems, they too require assumptions on the data such as the *restricted isometry property* for recovering true solutions [18, §7.5]. Furthermore, to converge to a local minimum, hard thresholding algorithms require the spectral norm of the measurement matrix to be less than one, which is a restrictive condition [5]. Besides hard thresholding algorithms, heuristics based on first-order algorithms such as the alternating direction method of multipliers (ADMM) have gained a lot of traction in the last few years. Though ADMM was originally designed to solve convex optimization problems, since the idea of implementing this algorithm as a general purpose heuristic to solve nonconvex optimization problems was introduced in [8, §9.1-9.2], ADMM-based heuristics have been applied successfully to approximately solve nonconvex problems in many different application areas. Takapoui *et al.* introduce an ADMM-based heuristic in [7] to solve mixed-integer

quadratic programming with applications in control and maximum-likelihood decoding problems. In [10], the authors extend the approach in [7] to a broader class of nonconvex problems, and they integrate the method into the widely used optimization modeling language `CVXPY` [16]. The biggest drawback of these heuristics comes from the fact that they take an algorithm designed to solve convex problems and apply it verbatim to a nonconvex setup. As a result, these algorithms often fail to converge, and even when they do, it need not be a local minimum, let alone a global one [9, §2.2]. Furthermore, empirical evidence suggests that the iterates of these algorithms may diverge even if they come arbitrarily close to a locally optimal solution during some iteration. In addition, even when the iterates do converge to a limit point, these heuristics cannot identify this limit point as a saddle point, local minimum, or neither [10, §9.1-9.2]. The main reason behind this inability is that these heuristics do not establish a clear relationship between the local minimum of ($\mathcal{P}$) and the fixed point set of the underlying operator that controls the iteration scheme. An alternative approach that has been quite successful empirically in finding low-rank solutions is to consider an unconstrained problem with Frobenius norm penalty and then using alternating minimization to compute a solution [59]. However, the alternating minimization approach may not converge to a solution and should be considered a heuristic [59, §2.4].

For these reasons above, in the last few years, there has been significant interest in addressing the nonconvexity present in many learning problems directly via a discrete optimization approach. In this way, a particular nonconvex optimization problem is formulated exactly using discrete optimization techniques and then specialized algorithms are developed to find a certifiably optimal solution. This approach has found considerable success in solving machine learning problems with sparse and low-rank optimization [52, 57]. A mixed integer optimization approach to compute near-optimal solutions for sparse regression problem, where $d = 1000$, is computed in [51]. In [54], the authors propose a cutting plane method for a similar problem, which works well with mild sample correlations and a sufficiently large dimension. In [53], the authors design and implement fast algorithms based on coordinate descent and local combinatorial optimization to solve sparse regression problem with a three-fold speedup where $d \approx 10^6$. In [55], the authors propose a framework for modeling and solving low-rank optimization problems to certifiable optimality via symmetric projection matrices. However, the runtime of these algorithms can often become prohibitively long as the problem dimensions grow [31]. Also, these discrete optimization algorithms have efficient implementations only for a narrow class of loss functions and constraint sets; they do not generalize well if a minor modification is made to the problem structure, and in such a case they often fail to find a solution point in a reasonable amount of time even for smaller dimensions [52]. Furthermore, one often relies on commercial softwares, such as `Gurobi`, `Mosek`, or `Cplex` to solve these discrete optimization problems, thus making the solution process somewhat opaque [31, 57].

1.3 Our approach

We propose an algorithm that alleviates the major drawbacks of aforementioned approaches. We propose the *nonconvex exterior-point optimization solver* (NExOS) algorithm: a novel first-order algorithm tailored for learning problems of the form ($\mathcal{P}$). The term *exterior-point* originates from the fact that the iterates approach a local minimum from outside of the feasible region; it is inspired by the convex exterior-point method first proposed by Fiacco and McCormick in the 1960s [46, §4]. The backbone of our approach is to address the nonconvexity by working with an asymptotically exact nonconvex penalization of ($\mathcal{P}$), which enjoys local convexity around local minima. In this

penalization, we replace the indicator function ι, the source of nonconvexity in ($\mathcal{P}$), with its *Moreau envelope* with positive parameter μ:

$$^{\mu}\iota(x) = \min_{y}\{\iota(y) + (1/2\mu)\|y - x\|^2\} = (1/2\mu)d^2(x), \tag{3}$$

where $d(x)$ is the Euclidean distance of the point x from the set $\mathcal{X}$. The main benefit of working with $^{\mu}\iota$ is that, it is (i) bounded and finite, (ii) jointly continuous in μ and x, (iii) a global underestimator of the indicator function ι, which improves with decreasing μ and becomes asymptotically equal to ι as μ approaches 0, and (iv) for any value of $\beta > 0$, the function $^{\mu}\iota + (\beta/2)\|\cdot\|^2$ is convex and differentiable on a neighborhood around local minima, where the size of the neighborhood is monotonically increasing with increase in μ. See [11, Proposition 12.9] for the first three properties, and Proposition 1 in §2 for the last one. The favorable features of $^{\mu}\iota$ motivate us to consider the following penalization formulation of ($\mathcal{P}$):

$$\text{minimize} \quad f(x) + \underbrace{(\beta/2)\|x\|^2 + {}^{\mu}\iota(x)}_{\equiv {}^{\mu}\mathbb{1}(x)}, \tag{$\mathcal{P}_\mu$}$$

where $x \in \mathbf{E}$ is the decision variable, and μ is a positive *penalty parameter*. We call the cost function in ($\mathcal{P}_\mu$) an *exterior-point minimization function*; the term is inspired by [46, §4.1]. The notation $^{\mu}\mathbb{1} \equiv {}^{\mu}\iota + (\beta/2)\|\cdot\|^2$ introduced in ($\mathcal{P}_\mu$) not only reduces notational clutter, but also alludes to a specific way of splitting the objective into two summands f and $^{\mu}\mathbb{1}$, which will ultimately allow us to establish convergence of our algorithm in §2. Because $^{\mu}\iota$ is an asymptotically exact approximation of ι as $\mu \to 0$, solving ($\mathcal{P}_\mu$) for a small enough value of the penalty parameter μ suffices for all practical purposes. Now that we have intuitively justified the exact penalization ($\mathcal{P}_\mu$), we are in a position to present our algorithm.

Algorithm 1: Nonconvex Exterior-point Optimization Solver (NExOS)

given: regularization parameter $\beta > 0$, an initial point z_{init}, initial penalty parameter μ_{init}, minimum penalty parameter $\mu_{\min}$, tolerance for the fixed point gap ϵ for each inner iteration, tolerance for stopping criterion δ for the outer iteration, and multiplicative factor $\rho \in (0, 1)$.

$\mu \coloneqq \mu_{\text{init}}$.
$z^0 \coloneqq z_{\text{init}}$.
Outer iteration. **while** *stopping criterion is not met* **do**
  Inner iteration. Using Algorithm 2, compute x_μ, y_μ, and z_μ that solve ($\mathcal{P}_\mu$) with tolerance ϵ, where $z_\mu^0 \coloneqq z^0$ is input as the initial point.
  Stopping criterion. **quit** *if* $|\big(f(\mathbf{\Pi}\, x_\mu) + (\beta/2)\|\mathbf{\Pi} x_\mu\|^2\big) - (f(x_\mu) + {}^{\mu}\mathbb{1}(x_\mu))| \leq \delta$.
  Set initial point for next inner iteration. $z^0 \coloneqq z_\mu$.
  Update μ. $\mu \coloneqq \rho\mu$.
return x_μ, y_μ, and z_μ

Algorithm 1 outlines NExOS. The main part is an outer loop that solves a sequence of penalized problems of the form ($\mathcal{P}_\mu$) with strictly decreasing penalty parameter μ, until the termination criterion is met, at which point the exterior-point minimization function is a sufficiently close approximation of the original cost function. For each μ, ($\mathcal{P}_\mu$) is solved by an inner algorithm,

denoted by Algorithm 2. One can derive Algorithm 2 by applying Douglas-Rachford splitting (DRS) [11, page 401] to ($\mathcal{P}_\mu$), this derivation is deferred to Appendix A.2. The term $\mathbf{prox}_{\gamma g}$ in Algorithm 2, which is the proximal operator of a function g (not necessarily convex) for $\gamma > 0$, is defined as:

$$\mathbf{prox}_{\gamma g}(x) = \underset{y \in \mathbf{E}}{\operatorname{argmin}} \left(g(y) + (1/2\gamma)\|y - x\|^2 \right), \tag{4}$$

which may be set-valued. However, for the convex function f, $\mathbf{prox}_{\gamma f}$ is always single-valued and computing it is equivalent to solving a convex optimization problem, which often can be done in closed form for many relevant cost functions in machine learning [14, pp. 449-450]. The notation $\mathbf{\Pi}(x)$ denotes the *projection operator* of x onto the constraint set $\mathcal{X}$, defined as

$$\mathbf{\Pi}(x) = \mathbf{prox}_{\gamma \iota}(x) = \operatorname{argmin}_{y \in \mathcal{X}} \left(\|y - x\|^2 \right). \tag{5}$$

As $\mathcal{X}$ is nonconvex, there can be multiple projections onto it from a point outside $\mathcal{X}$, hence, $\mathbf{\Pi}$ can be set-valued. However, NExOS is designed to ensure that $\mathbf{\Pi}$ is single-valued at our iterates.

Algorithm 2: Inner Algorithm for ($\mathcal{P}_\mu$).

given: starting point z^0, tolerance for the fixed point gap ϵ, and proximal parameter $\gamma > 0$.

$n \coloneqq 0$.
$\kappa \coloneqq \frac{1}{\beta\gamma+1}$.
$\theta \coloneqq \frac{\mu}{\gamma\kappa+\mu}$.
while $\|x^n - y^n\| > \epsilon$ **do**
  Compute $x^{n+1} \coloneqq \mathbf{prox}_{\gamma f}(z^n)$.
  Compute $\tilde{y}^{n+1} \coloneqq \kappa\left(2x^{n+1} - z^n\right)$.
  Compute $y^{n+1} \coloneqq \theta\tilde{y}^{n+1} + (1-\theta)\,\mathbf{\Pi}\left(\tilde{y}^{n+1}\right)$.
  Compute $z^{n+1} \coloneqq z^n + y^{n+1} - x^{n+1}$.
  Update $n \coloneqq n + 1$.
return x^n, y^n, and z^n.

In principle, we could start with a very small value of μ in ($\mathcal{P}_\mu$) and then focus on finding an approximate solution to ($\mathcal{P}$). However, in practice, such a one-shot approach may be problematic for the following reasons. In our convergence analysis, we will show that as long as the initial point for the inner algorithm lies in the region where $\mu_{\mathbb{I}}$ is convex and smooth, the inner algorithm will linearly converge to a locally optimal solution to ($\mathcal{P}_\mu$). However, if we start NExOS with a very small μ, then finding an initial point that lies in this favorable region can become be as hard as solving the original problem. This issue can be addressed by executing a sequence of inner algorithms, with each inner algorithm (Algorithm 2) solving ($\mathcal{P}_\mu$) for a fixed value of μ. We will start the first inner algorithm with a relatively large value of μ and then gradually decrease μ until a certain termination criterion is met. With the initial value of μ, ($\mathcal{P}_\mu$) is not a good approximation of the original problem, but it is easier to find a locally optimal solution to ($\mathcal{P}_\mu$). In §2, we show that for a reasonable reduction in μ, ($\mathcal{P}_\mu$) becomes an increasingly accurate approximation of ($\mathcal{P}$), and the solution found by the previous value of the penalty parameter will stay in the new region of convexity and smoothness, thus acting as a good initial point for the new inner algorithm. This allows for linear convergence of the inner algorithm to the solution for the new value of the penalty parameter.

Here we note that, conceptually, our framework is similar to sequential unconstrained minimization technique (SUMT) [46]. However, the reason why our method requires a sequence of inner algorithms is fundamentally different than SUMT. An algorithm based on SUMT, *e.g.*, the interior point method, requires multiple inner algorithms to deal with convex inequality constraints present in the problem and to avoid a situation where the Hessian varies rapidly near the boundary of the constraint set [15, page 564]. On the other hand, NExOS, which is designed for nonconvex problems, needs the inner algorithms to keep the iterates in the regions of convexity and smoothness in order to ensure linear convergence.

1.4 Contributions

Our main contributions are as follows.

1. The main contribution of this work is to propose NExOS: a novel first-order algorithm tailored for learning problems with convex cost functions over nonconvex constraint sets. We prove that NExOS converges to a locally optimal solution under mild technical conditions without requiring any stringent assumption on the data unlike the convex relaxation or hard threhsholding based approaches. In comparison with the discrete optimization approach, we demonstrate that NExOS remains computationally tractable as the problem dimension grows, is generalizable to a much broader class of loss functions and constraint sets, and is not dependent on any commercial software for efficient implementation. Additionally, we illustrate that our method, being fast and scalable with theoretical guarantees, can be used to provide a warm-start point to the discrete optimization algorithms, thus complementing them.
2. We prove that NExOS, besides avoiding the drawbacks of convex relaxation and discrete optimization approach, has the following desirable characteristics. First, the penalized problem has strongly convexity and smoothness around local minima, but can be made arbitrarily close to the original nonconvex problem by reducing only one penalty parameter. Second, the inner algorithm finds local minima for the penalized problems at a linear convergence rate, and as the penalty parameter goes to zero, the local minima of the penalized problems converge to a local minimum of the original problem.
3. We implement NExOS in the open-source `Julia` package `NExOS.jl` and test it extensively on many instances of different nonconvex learning problems that fit our setup with an emphasis on sparse and low-rank optimization problems. We find that in spite of being a general purpose solver, `NExOS.jl` very quickly computes solutions that are either competitive or better in comparison with specialized algorithms on various performance measures. For example, we empirically show that for the sparse regression problem, NExOS finds solutions with 4% higher in support recovery (recovers 98% of the true signal on average) but 11 times smaller in training loss in comparison with Lasso. Furthermore, when compared with globally optimal solutions to sparse regression problems, a parallel implementation our algorithm provides near-optimal solutions (99% optimal on average) in less than 3 seconds with a runtime that increases linearly, where the runtime of the state-of-the art global solver `Gurobi` grows exponentially. For low-rank factor analysis, NExOS provides solutions with 3 times smaller training loss but 2 times higher *proportion of explained variance* (a measure of how much of the data is explained by the solution) on average compared to the nuclear norm method. `NExOS.jl` is available at

   https://github.com/Shuvomoy/NExOS.jl.

Organization of the paper. The rest of the paper is organized as follows. We provide convergence analysis of the algorithm in §2. Then we demonstrate the performance of our algorithm on several nonconvex machine learning problems of significant current interest in §3. The concluding remarks are presented in §4.

Notation and notions. Domain of a function $h : \mathbf{E} \to \mathbf{R} \cup \{\infty\}$ is defined as $\mathbf{dom}\, h = \{x \in \mathbf{E} \mid h(x) < \infty\}$. A function f is proper if its domain is nonempty, and it is lower-semicontinuous if its epigraph $\mathbf{epi}\, f = \{(x,t) \in \mathbf{E} \times \mathbf{R} \mid f(x) \leq t\}$ is a closed set. By $B(x;r)$ and $\overline{B}(x;r)$, we denote an open ball and a closed ball of radius r and center x, respectively.

A set-valued operator $\mathbb{A} : \mathbf{E} \rightrightarrows \mathbf{E}$ maps an element x in $\mathbf{E}$ to a set $\mathbb{A}(x)$ in $\mathbf{E}$; its domain is defined as $\mathbf{dom}\, \mathbb{A} = \{x \in \mathbf{E} \mid \mathbb{A}(x) \neq \emptyset\}$, its range is defined as $\mathbf{ran}\, \mathbb{A} = \bigcup_{x \in \mathbf{E}} \mathbb{A}(x)$, and it is completely characterized by its graph: $\mathbf{gra}\, \mathbb{A} = \{(u,x) \in \mathbf{E} \times \mathbf{E} \mid u \in \mathbb{A}(x)\}$. Furthermore, we define $\mathbf{fix}\, \mathbb{A} = \{x \in \mathbf{E} \mid x \in \mathbb{A}(x)\}$, and $\mathbf{zer}\, \mathbb{A} = \{x \in \mathbf{E} \mid 0 \in \mathbb{A}(x)\}$. For every x, addition of two operators $\mathbb{A}_1, \mathbb{A}_2 : \mathbf{E} \rightrightarrows \mathbf{E}$, denoted by $\mathbb{A}_1 + \mathbb{A}_2$, is defined as $(\mathbb{A}_1 + \mathbb{A}_2)(x) = \mathbb{A}_1(x) + \mathbb{A}_2(x)$, subtraction is defined analogously, and composition of these operators, denoted by $\mathbb{A}_1\mathbb{A}_2$, is defined as $\mathbb{A}_1\mathbb{A}_2(x) = \mathbb{A}_1(\mathbb{A}_2(x))$; note that order matters for composition. Also, if $\mathcal{S} \subseteq \mathbf{E}$ is a nonempty set, then $\mathbb{A}(\mathcal{S}) = \cup\{\mathbb{A}(x) \mid x \in \mathcal{S}\}$.

An operator $\mathbb{A} : \mathbf{E} \to \mathbf{E}$ is nonexpansive on some set $\mathcal{S}$ if it is Lipschitz continuous with Lipschitz constant 1 on $\mathcal{S}$. On the other hand, $\mathbb{A}$ is firmly nonexpansive on $\mathcal{S}$ if and only if its reflection operator $2\mathbb{A} - \mathbb{I}$ is nonexpansive on $\mathcal{S}$. A firmly nonexpansive operator is always nonexpansive [11, page 59].

2 Convergence analysis

This section is organized as follows. We start with the definition of the local minima, followed by the assumptions we use in our convergence analysis. Then, we discuss the convergence roadmap, where the first step involves showing that the exterior point minimization function is locally strongly convex and smooth around local minima, and the second step entails connecting the local minima with the underlying operator controlling NExOS. Finally, we present the main convergence result.

We start with the definition of local minimum for of ($\mathcal{P}$). Recall that, according to our setup the set $\mathcal{X}$ is prox-regular at local minimum.

Definition 2 (Local minimum of ($\mathcal{P}$))**.** A point $\bar{x} \in \mathcal{X}$ is a local minimum of ($\mathcal{P}$) if the set $\mathcal{X}$ is prox-regular at $\bar{x}$, and there exists a closed ball $\overline{B}(\bar{x};r)$ such that for all $y \in \mathcal{X} \cap \overline{B}(\bar{x};r) \setminus \{\bar{x}\}$, we have

$$f(\bar{x}) + (\beta/2)\|\bar{x}\|^2 < f(y) + (\beta/2)\|y\|^2.$$

In the definition above, the strict inequality is due to the strongly convex nature of the objective $f + (\beta/2)\|\cdot\|^2$ and follows from [49, Proposition 2.1] and [12, Theorem 6.12].

Next, we state and justify the assumptions used in our convergence analysis.

Assumption 1 (Strong convexity and smoothness of f)**.** *The function f in ($\mathcal{P}_\mu$) is α-strongly convex and L-smooth where $L > \alpha > 0$, i.e., $f - (\alpha/2)\|\cdot\|^2$ is convex and $f - (L/2)\|\cdot\|^2$ is concave.*

Assumption 2 (Problem ($\mathcal{P}$) is not trivial)**.** *The unique solution to the unconstrained strongly convex problem*

$$\text{minimize} \quad f(x) + (\beta/2)\|x\|^2 \ , \tag{6}$$

does not lie in $\mathcal{X}$.

Assumption 1 is considered a standard assumption in proving convergence results in machine learning [18]. The L-smoothness in f is equivalent to its gradient ∇f being $L-$Lipschitz everywhere on $\mathbf{E}$ [11, Theorem 18.15]. This assumption helps us in establishing linear convergence of the inner algorithms of NExOS. For the case where f is does not satisfy Assumption 1, we provide a minor modification of NExOS with same convergence guarantee that works with a strongly convex, smooth, and arbitrarily close approximation of the function (see Appendix B.1).

Assumption 2 imposes that a local minimum of ($\mathcal{P}$) is not the global minimum of its unconstrained convex relaxation, which does not incur any loss of generality. We can solve the unconstrained strongly convex optimization problem (6) and check if the corresponding minimizer lies in $\mathcal{X}$; if that is the case, then that minimizer is also the global minimizer of ($\mathcal{P}$), and there is no point in solving the nonconvex problem. As this can be easily checked by solving an unconstrained convex optimization problem (6), imposing Assumption 2 does not cause any loss of generality.

We next discuss our convergence roadmap. Convergence of NExOS is controlled by the following operator that we can construct by putting the first two iterates of Algorithm 2 in the third iterate and is called the DRS operator of ($\mathcal{P}_\mu$):

$$\mathbb{T}_\mu = \mathbf{prox}_{\gamma \mu_{\emptyset}}\left(2\mathbf{prox}_{\gamma f} - \mathbb{I}\right) + \mathbb{I} - \mathbf{prox}_{\gamma f}, \tag{7}$$

where $\mu > 0$, and $\mathbb{I}$ stands for the identity operator in $\mathbf{E}$, *i.e.*, for any $x \in \mathbf{E}$, we have $\mathbb{I}(x) = x$. Using $\mathbb{T}_\mu$, the inner algorithm—Algorithm 2—can be written very compactly in a single line as

$$z^{n+1} = \mathbb{T}_\mu\left(z^n\right) \tag{$\mathcal{A}_\mu$}$$

where μ is the penalty parameter and the iterate z^n is initialized at the fixed point found in the previous inner algorithm.

To show the convergence of NExOS, we first show that for some $\mu_{\max} > 0$, for any $\mu \in (0, \mu_{\max}]$, the exterior point minimization function $f + \mu_{\emptyset}$ is strongly convex and smooth on some open ball $B(\bar{x}; r_{\max})$, where it will attain a unique local minimum x_μ. Then we show that for $\mu \in (0, \mu_{\max}]$, the operator $\mathbb{T}_\mu(x)$ will be contractive in x and Lipschitz continuous in μ, and connects its fixed point set $\mathbf{fix}\,\mathbb{T}_\mu$ with the local minima x_μ, via the relationship $x_\mu = \mathbf{prox}_{\gamma f}(\mathbf{fix}\,\mathbb{T}_\mu)$. Finally, in the main convergence result, we show that for a sequence of penalty parameters $\mathfrak{M} = \{\mu_1, \mu_2, \mu_3, \ldots, \mu_N\}$, if we apply NExOS to $\mathfrak{M}$, then for all $\mu_m \in \mathfrak{M}$,the inner algorithm will linearly converge to x_{μ_m}, and as $\mu_N \to 0$, we will have $x_{\mu_N} \to \bar{x}$.

We next present a proposition that shows that the exterior point minimization function in ($\mathcal{P}_\mu$) will be locally strongly convex and smooth around local minima for our selection of penalty parameters, even though ($\mathcal{P}$) is nonconvex. Furthermore, as the penalty parameter goes to zero, the local minimum of ($\mathcal{P}_\mu$) converges to the local minimum of the original problem ($\mathcal{P}$). As a consequence of this result, under proper initialization, NExOS would be able to solve the sequence of penalized problems $\{\mathcal{P}_\mu\}_{\mu \in (0, \mu_{\text{init}}]}$ similar to convex optimization problems; we will prove this in our main convergence result (Theorem 1).

Proposition 1 (Attainment of local minimum by $f + \mu\mathbb{0}$)**.** *Let $\bar{x}$ be a local minimum to ($\mathcal{P}$), where Assumptions 1 and 2 hold. Then the following hold.*

(i) There exist $\mu_{\max} > 0$ and $r_{\max} > 0$ such that for any $\mu \in (0, \mu_{\max}]$, the exterior point minimization function $f + \mu\mathbb{0}$ in ($\mathcal{P}_\mu$) will be strongly convex and smooth in the open ball $B(\bar{x}; r_{\max})$ and will attain a unique local minimum x_μ in this ball.

(ii) As $\mu \to 0$, this local minimum x_μ will go to $\bar{x}$ in limit, i.e., $x_\mu \to \bar{x}$.

Proof. See Appendix B.2. □

Remark 1 (Selecting μ_{init}). Using Proposition 1 we can select and μ_{init} for our convergence analysis as follows: $\mu_{\text{init}} \leq \mu_{\max}$.

Because the exterior point minimization function is locally strongly convex and smooth, intuition suggests that locally the DRS operator of ($\mathcal{P}_\mu$) would behave in a manner analgous to that of a DRS operator of a composite convex optimization problem. Recall that, when we minimize a sum of two convex functions where one of them is strongly convex and smooth, the corresponding DRS operator is contractive [17, Theorem 1]. So, we can expect that the DRS operator for ($\mathcal{P}_\mu$) would be locally contractive around a local minimum, which indeed turns out to be the case as proven in the next proposition. Furthermore, the next proposition shows that $\mathbb{T}_\mu(x)$ is locally Lipschitz continuous in the penalty parameter μ around a local minimum for fixed x. As $\mathbb{T}_\mu(x)$ is locally contractive in x and Lipschitz continuous in μ, it ensures that as we reduce the penalty parameter μ, the local minimum x_μ of ($\mathcal{P}_\mu$) found by NExOS does not change abruptly.

Proposition 2 (Characterization of $\mathbb{T}_\mu$)**.** *Let $\bar{x}$ be a local minimum to ($\mathcal{P}$), where Assumptions 1 and 2 hold. Then the following hold.*

(i) For any $\mu \in (0, \mu_{\max}]$, the operator $\mathbb{T}_\mu$ is κ'-contractive, i.e., there exists a contraction factor $\kappa' \in (0, 1)$ such that for any $x_1, x_2 \in B(\bar{x}; r_{\max})$ and $\mu \in (0, \mu_{\max}]$, we have

$$\|\mathbb{T}_\mu(x_1) - \mathbb{T}_\mu(x_2)\| \leq \kappa' \|x_1 - x_2\|.$$

(ii) For any $x \in B(\bar{x}; r_{\max})$, the operator $\mathbb{T}_\mu(x)$ is Lipschitz continuous in μ, i.e., there exists an $\ell > 0$ such that for any $\mu_1, \mu_2 \in (0, \mu_{\max}]$ and $x \in B(\bar{x}; r_{\max})$, we have

$$\|\mathbb{T}_{\mu_1}(x) - \mathbb{T}_{\mu_2}(x)\| \leq \ell \|\mu_1 - \mu_2\|.$$

Proof. See Appendix B.3. □

If ($\mathcal{A}_\mu$) converges to a point z_μ, then z_μ would be a fixed point of the DRS operator $\mathbb{T}_\mu$. Establishing the convergence of NExOS necessitates connecting the local minimum x_μ of ($\mathcal{P}_\mu$) to the fixed point set of $\mathbb{T}_\mu$, which is achieved by the next proposition. Because our DRS operator locally behaves in a manner similar to the DRS operator of a convex optimization problem as shown by Proposition 2, it is natural to expect that the connection between x_μ and z_μ in our setup would be similar to that of a convex setup, but in a local sense. This indeed turns out to be the case as proven in the next proposition. The statement of this proposition is structurally similar to [11, Proposition 25.1(ii)] that establishes a similar relationship globally for a convex setup, whereas our result is established around the local minima of ($\mathcal{P}_\mu$).

Proposition 3 (Relationship between local minima of ($\mathcal{P}$) and $\mathbf{fix}\,\mathbb{T}_\mu$)**.** *Let $\bar{x}$ be a local minimum to ($\mathcal{P}$), where Assumptions 1 and 2 hold and $\mu \in (0, \mu_{\max}]$. Then,*

$$x_\mu = \operatorname*{argmin}_{B(\bar{x};r_{\max})} f(x) + {}^{\mu}\mathbb{0}(x) = \mathbf{prox}_{\gamma f}\left(\mathbf{fix}\,\mathbb{T}_\mu\right), \tag{8}$$

where the sets $\mathbf{fix}\,\mathbb{T}_\mu$, and $\mathbf{prox}_{\gamma f}\left(\mathbf{fix}\,\mathbb{T}_\mu\right)$ are singletons over $B(\bar{x}; r_{\max})$.

Proof. See Appendix B.4. □

Before we present the main convergence result, we provide a helper lemma, which shows how the distances between x_μ, z_μ and $\bar{x}$ change as μ is varied in Algorithm 1. Additionally, this lemma provides the range for the proximal parameter γ. If $\mathcal{X}$ is a bounded set satisfying $\|x\| \leq D$ for all $x \in \mathcal{X}$, then term $\max_{x \in B(\bar{x};r_{\max})} \|\nabla f(x)\|$ in this lemma can be replaced with $L \times D$.

Lemma 1 (Distance between local minima of ($\mathcal{P}$) with local minima of of ($\mathcal{P}_\mu$))**.** *Let $\bar{x}$ be a local minimum to ($\mathcal{P}$) over $B(\bar{x}; r_{\max})$, where Assumptions 1 and 2 hold. Then the following hold.*

(i) For any $\mu \in (0, \mu_{\max}]$, the unique local minimum x_μ of ($\mathcal{P}_\mu$) over $B(\bar{x}; r_{\max})$ satisfies

$$\|x_\mu - \bar{x}\| < \frac{1}{\eta'} r_{\max}, \tag{9}$$

for some $\eta' > 1$.

(ii) Let z_μ be the unique fixed point of $\mathbb{T}_\mu$ over $B(\bar{x}; r_{\max})$ corresponding to x_μ. Then for any $\mu \in (0, \mu_{\max}]$, we have

$$r_{\max} - \|x_\mu - \bar{x}\| > \frac{\eta' - 1}{\eta'} r_{\max}, \tag{10}$$

and

$$r_{\max} - \|z_\mu - \bar{x}\| > \psi, \tag{11}$$

where

$$\psi = \frac{\eta' - 1}{\eta'} r_{\max} - \gamma \max_{x \in B(\bar{x};r_{\max})} \|\nabla f(x)\| > 0 \tag{12}$$

with the proximal parameter γ taken to satisfy

$$0 < \gamma < \frac{\eta' - 1}{\eta'} \frac{r_{\max}}{\max_{x \in B(\bar{x};r_{\max})} \|\nabla f(x)\|}. \tag{13}$$

Furthermore,

$$\min_{\mu \in (0, \mu_{\max}]} \{(r_{\max} - \|z_\mu - \bar{x}\|) - \psi\} > 0. \tag{14}$$

Proof. See Appendix B.5. □

We now present our main convergence results for NExOS. For convenience, we denote the n-th iterates of the inner algorithm of NExOS for penalty parameter μ by $\{x_\mu^n, y_\mu^n, z_\mu^n\}$. In the theorem, an ϵ-approximate fixed point $\widetilde{z}$ of $\mathbb{T}_\mu$ is defined by $\|\widetilde{z} - \mathbb{T}_\mu(\widetilde{z})\| \leq \epsilon$. Furthermore, define:

$$\bar{\epsilon} := \min\left\{\frac{1}{2}\min_{\mu\in(0,\mu_{\max}]}\left\{(r_{\max} - \|z_\mu - \bar{x}\|) - \psi\right\}, (1-\kappa')\psi\right\} > 0, \tag{15}$$

where $\kappa' \in (0,1)$ is the contraction factor of $\mathbb{T}_\mu$ for any $\mu > 0$ (cf. Proposition 2) and the right-hand side is positive due to (14), (12). Theorem 1 states that if we have a good initial point z_{init} for the first penalty parameter μ_{init}, then NExOS will construct a finite sequence of penalty parameters such that all the inner algorithms for these penalty parameters will linearly converge to solutions of the corresponding inner problems.

Theorem 1 (Convergence result for NExOS)**.** *Let $\bar{x}$ be a local minimum to (P), where Assumptions 1 and 2 hold. Suppose that the fixed-point tolerance ϵ for Algorithm 2 satisfies $\epsilon \in [0, \bar{\epsilon})$, where $\bar{\epsilon}$ is defined in (15). The proximal parameter γ is selected to satisfy (13). In this setup, NExOS will construct a finite sequence of strictly decreasing penalty parameters $\mathfrak{M} = \{\mu_1 := \mu_{\text{init}}, \mu_2 = \rho\mu_1, \mu_3 = \rho\mu_2, \ldots\}$, with $\mu_{\text{init}} \leq \mu_{\max}$ and $\rho \in (0,1)$, such that we have the following recursive convergence property. For any $\mu \in \mathcal{M}$, if an ϵ-approximate fixed point of $\mathbb{T}_\mu$ over $B(\bar{x}; r_{\max})$ is used to initialize the inner algorithm for penalty parameter $\rho\mu$, then the corresponding inner algorithm iterates $z_{\rho\mu}^n$ linearly converges to $z_{\rho\mu}$ that is the unique fixed point of $\mathbb{T}_{\rho\mu}$ over $B(\bar{x}, r_{\max})$, and the iterates $x_{\rho\mu}^n, y_{\rho\mu}^n$ linearly converge to $x_{\rho\mu} = \mathbf{prox}_{\gamma f}(z_{\rho\mu})$, which is the unique local minimum to $(\mathcal{P}_{\rho\mu})$ over $B(\bar{x}; r_{\max})$).*

Proof. See Appendix B.6. □

From Theorem 1, we see that an ϵ-approximate fixed point of $\mathbb{T}_{\rho\mu}$ over $B(\bar{x}; r_{\max})$ can be computed and then used to initialize the next inner algorithm for penalty parameter $\rho^2\mu$; this chain of logic makes each inner algorithm converge to the corresponding locally optimal solution. Finally, for the convergence of the first inner algorithm we have the following result, which states that if the initial point z_{init} is not too far away from $B(\bar{x}; r_{\max})$, then the first inner algorithm of NExOS for penalty parameter μ_1 converges to a locally optimal solution of $(\mathcal{P}_{\mu_1})$.

Lemma 2 (Convergence of the first inner algorithm)**.** *Let $\bar{x}$ be a local minimum to (P), where Assumptions 1 and 2 hold. Let z_{init} be the chosen initial point for $\mu_1 := \mu_{\text{init}}$ such that $\overline{B}(z_{\mu_1}; \|z_{\text{init}} - z_{\mu_1}\|) \subseteq B(\bar{x}; r_{\max})$, where z_{μ_1} be the corresponding unique fixed point of $\mathbb{T}_{\mu_1}$. Then, $z_{\mu_1}^n$ linearly converges to z_{μ_1} and both $x_{\mu_1}^n$ and $y_{\mu_1}^n$ linearly converge to the unique local minimum x_{μ_1} of $(\mathcal{P}_{\mu_1})$ over $B(\bar{x}; r_{\max})$.*

Proof. See Appendix B.7. □

3 Numerical experiments

In this section, we apply NExOS to the following nonconvex machine learning problems of substantial current interest for both synthetic and real-world datasets: regressor selection problem in §3.1, affine rank minimization problem in §3.2, and low-rank factor analysis problem in §3.3. We illustrate

that NExOS produces solutions that are either competitive or better in comparison with the other approaches on different performance measures. We have implemented NExOS in `NExOS.jl` solver, which is an open-source software package written in the `Julia` programming language. `NExOS.jl` can address any optimization problem of the form ($\mathcal{P}$). The code and documentation are available online at:

https://github.com/Shuvomoy/NExOS.jl.

To compute the proximal operator of a function f that has closed form or easy-to-compute solution, `NExOS.jl` uses the open-source `Julia` package `ProximalOperators.jl` [47]. When f is a constrained convex function (*i.e.*, a convex function over some convex constraint set) with no closed form proximal map, `NExOS.jl` computes the proximal operator by using the open-source `Julia` package `JuMP` [21] and any of the commercial or open-source solver supported by it. The set $\mathcal{X}$ can be any prox-regular nonconvex set fitting our setup. Our implementation is readily extensible using `Julia` abstract types so that the user can add support for additional convex functions and prox-regular sets.

The numerical study is executed on a computer with Intel Core i5-8250U CPU with 8 GB RAM running Windows 10 Pro operating system. The datasets considered in this section, unless specified otherwise, are available online at:

https://github.com/Shuvomoy/NExOS_Numerical_Experiments_Datasets.

In applying NExOS, we use the following values. We take the starting value of μ to be 2, and reduce this value with a multiplicative factor of 0.5 during each iteration of the outer loop until the termination criterion is met. The value of the proximal parameter γ is chosen to be 10^{-3}. We initialize our iterates at $\mathbf{0}$. Maximum number of inner iterations for a fixed value of μ is taken to be 1000. The tolerance for the fixed point gap is taken to be 10^{-4} and the tolerance for the termination criterion is taken to be 10^{-6}. Value of β is taken to be 10^{-8}.

3.1 Regressor selection

In (SR), we set $\mathcal{X} := \{x \mid \|x\|_\infty \leq \Gamma, \mathbf{card}(x) \leq k\}$, and $f(x) := \|Ax - b\|_2^2$. A projection onto $\mathcal{X}$ can be computed using the formula in [18, §2.2], whereas the proximal operator for f can be computed using the formula in [20, §6.1.1]. Now we are in a position to apply NExOS to this problem.

3.1.1 Synthetic dataset: comparison with Lasso

We compare the solution found by NExOS with the solution found by an algorithm based on Lasso (least absolute shrinkage and selection operator).

Lasso. Lasso is perhaps the most well-known method for solving the regressor selection problem, that computes an approximate solution as follows. First, Lasso solves:

$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda\|x\|_1 + (\beta/2)\|x\|_2^2, \tag{16}$$

where λ is a parameter that is related to the sparsity of the decision variable $x \in \mathbf{R}^d$. To solve (16), we have used glmnet, which is one of the most popular packages to implement Lasso [1, pp. 50-52].

To compute the value of λ corresponding to a desired k such that $\mathbf{card}(x) \leq k$ we follow the method proposed in [6, §3.4] and [15, Example 6.4]. We solve the problem (16) for different values of λ, and find the smallest value of λ for which we have $\mathbf{card}(x) \leq k$, and we consider the sparsity pattern of the corresponding solution $\tilde{x}$. Let the index set of zero elements of $\tilde{x}$ be $\mathcal{Z}$, where $\mathcal{Z}$ has $d-k$ elements. Then the Lasso-based algorithm solves the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2^2 + (\beta/2)\|x\|_2^2 \\ \text{subject to} & (\forall j \in \mathcal{Z}) \quad x_j = 0, \end{array} \tag{17}$$

where $x \in \mathbf{R}^d$ is the decision variable. To solve this problem, we have used Gurobi's quadratic optimization solver.

Data generation process and setup. The data generation procedure is similar to [10]. In this numerical experiment, we vary m from 50 to 150, and for each value of m, we generate 50 random instances. For a certain value of m, the matrix $A \in \mathbf{R}^{m \times 2m}$ is generated from an independent and identically distributed normal distribution with $\mathcal{N}(0,1)$ entries. We choose $b = A\widetilde{x} + v$, where $\widetilde{x}$ is drawn uniformly from the set of vectors satisfying $\mathbf{card}(\widetilde{x}) \leq \lfloor \frac{m}{5} \rceil$ and $\|\widetilde{x}\|_\infty \leq 1$. The vector v corresponds to noise, and is drawn from the distribution $\mathcal{N}\left(0, \sigma^2 I\right)$, where $\sigma^2 = \|A\widetilde{x}\|_2^2/(400/m)$, which keeps the signal-to-noise ratio to approximately 20.

Results. The results displayed in the figures are averaged over 50 simulations for each value of m, and also show one-standard-error bands that represent one standard deviation confidence interval around the mean.

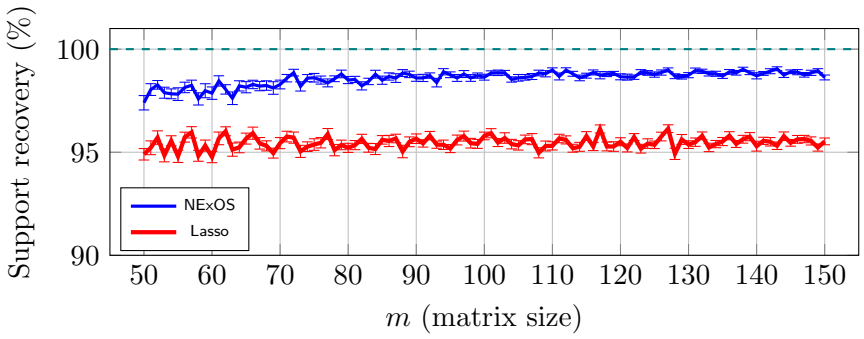
Figure 1a shows the support recovery (%) of the solutions found by NExOS and Lasso. Given a solution $x \in \mathbf{R}^d$ and true signal $x^{\text{True}} \in \mathbf{R}^d$, the support recovery (%) is defined as

$$100 \times \frac{\sum_{i=1}^{d} 1_{\{\text{sign}(x_i) = \text{sign}(x_i^{\text{True}})\}}}{d},$$

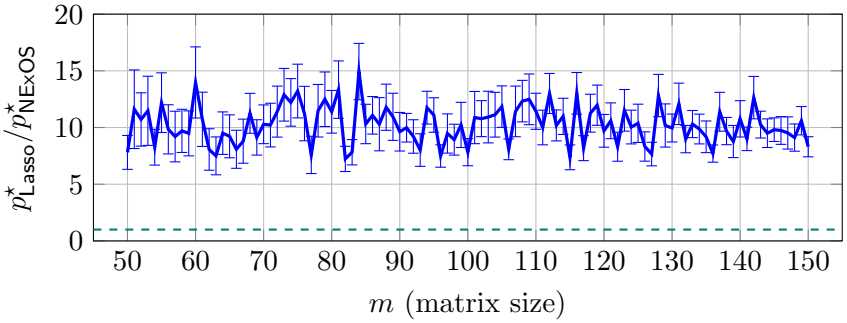
where $1_{\{\cdot\}}$ evaluates to one if $(\cdot)$ is true and 0 else, and $\text{sign}(t)$ is 1 for $t > 0$, -1 for $t < 0$, and 0 for $t = 0$. So, higher the support recovery, better is the quality of the found solution. We see that NExOS always recovers most of the original signal's support, and it does it better than Lasso consistently. On average, NExOS recovers 4% more of the support than Lasso.

Figure 1b compares the quality of the solutions found by both algorithms in terms of training loss. We see that the training loss of the solution found by Lasso is much larger than the objective value found by NExOS in all the cases averaged over all the 50 instances for each size of the problem.

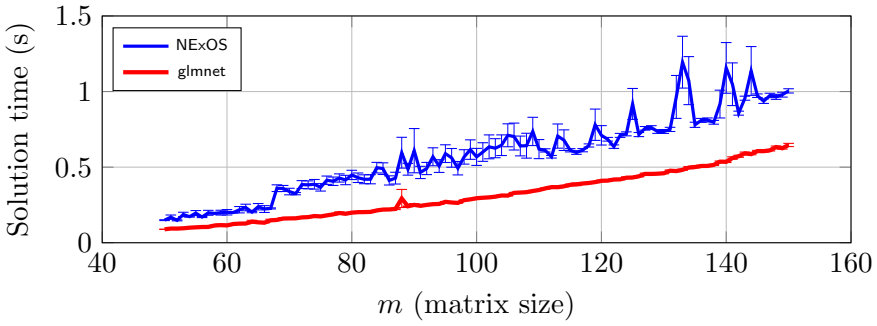
Finally, in Figure 1c, we compare the solution times of NExOS to that of glmnet. Note that, to compare only with glmnet, we have only considered the solution time to compute one solution to (16) by glmnet and have excluded the solution time to solve (17).We see that glmnet is slightly faster than Algorithm 1. This slower performance is due to the fact that our algorithm is a general purpose method for nonconvex optimization problems applicable to any convex loss function over a prox-regular constraint set, whereas glmnet is specifically optimized for the convexified sparse regression problem with a very specific objective function. Furthermore, by choosing a slightly larger value of termination tolerance δ, we can make our algorithm as fast as glmnet, though the precision of the solution may not be very high.



(a) Support recovery (%) by Lasso and NExOS vs m



(b) Ratio of training loss of the solution found by Lasso and NExOS vs m



(c) Solution time by Lasso and NExOS vs m

Figure 1: Sparse regression problem: comparison between NExOS and Lasso

Here we note that, if we consider the total solution time to solve both (16) and (17), then NExOS is faster; but we omit this timing result as Gurobi is not in particular optimized to solve (17) and increases the solution time disproportionately.

To summarize, the experiments with the synthetic datasets suggest that the solution found by NExOS is consistently of higher quality in terms of support recovery and training loss than that of Lasso.

3.1.2 Synthetic dataset: comparison with Gurobi

The next experiment we perform is to see how well NExOS does in getting close to the absolute optimal value of the original problem. NExOS is guaranteed to provide a locally optimal solution under regularity conditions; however, if we are interested in reducing the objective value as much as possible to investigate how close we get to the absolute minimum value, then it is reasonable to initialize NExOS for different random points and take the solution associated with the least objective value.

Formulation for Gurobi. The sparse regression problem (SR) can be modeled as the following mixed integer quadratic optimization problem:

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2^2 + (\beta/2)\|x\|_2^2 \\ \text{subject to} & |x_i| \leq \Gamma y_i, \qquad i = 1, \ldots, d \\ & \sum_{i=1}^d y_i \leq k \\ & x \in \mathbf{R}^d, y \in \{0,1\}^d, \end{array}$$

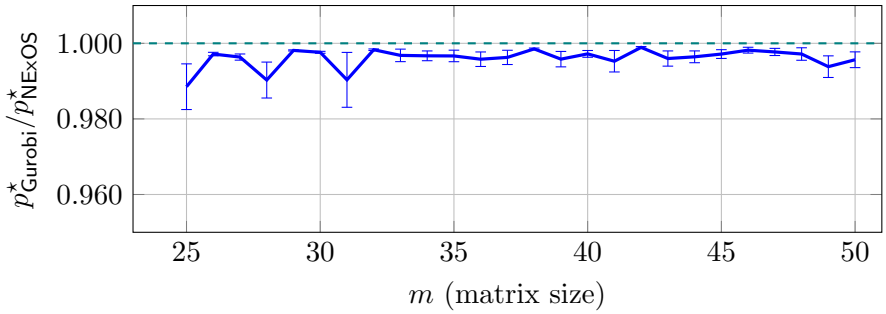
where Γ is a large number that can be chosen based on the problem data [31]. This mixed integer quadratic optimization problem can be solved exactly using Gurobi.

Data generation process and setup. We vary m from 25 to 50, and for each value of m, we generate 50 instances like before. We limit the size of the problems because the solution time by Gurobi becomes too large for comparison if we go beyond the aforesaid size. For each of the instances, we generate 100 random instances from the uniform distribution over the interval $[-\Gamma, \Gamma]$. We speed up the calculation by running NExOS for different initializations by using multi-threading capability in Julia, each thread running NExOS for a different random initialization parallelly. Number of threads for this experiment was 4.

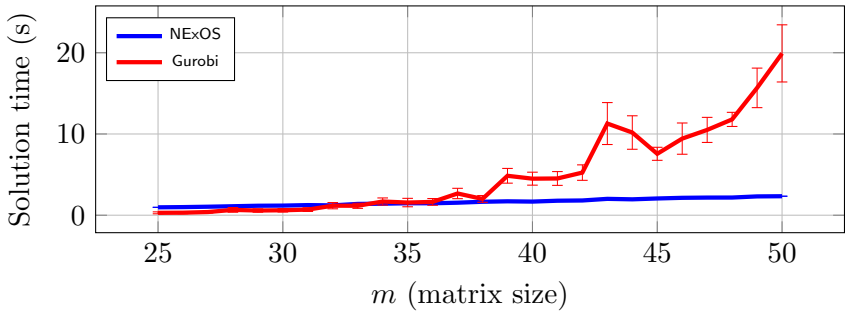
Results. The results displayed in Figure 2a and Figure 2b are averaged over 50 simulations for each value of m, and also show one-standard-error bands.

Figure 2a shows the ratio between the objective value found by NExOS and Gurobi. The closer it is to 1, the better is the quality of the solution found by NExOS in terms of minimizing the objective value. We see that in most cases NExOS is able to find a solution that is very close to the optimal solution.

In Figure 2b, we compare the solution times of NExOS to that of Gurobi. For smaller problems, Gurobi is somewhat faster than NExOS, however once we go beyond $m \geq 34$, the solution time by Gurobi starts to increase in an exponential fashion whereas NExOS scales linearly. Beyond $m \geq 50$, comparing the runtimes is not meaningful as Gurobi cannot find a solution in 2 minutes.



(a) Ratio of objective value found by Gurobi and NExOS vs m



(b) Solution time by Gurobi and NExOS vs m

Figure 2: Sparse regression problem: comparison between NExOS and Gurobi

3.1.3 Experiments and results for real-world dataset

Description of the dataset. To investigate the performance of our algorithm on a real-world dataset, we consider the problem of murder rate detection per 100,000 people comprising of $m = 2215$ communities in the United States with $d = 101$ attributes. By preprocessing the data, we remove some of the columns with missing entries, and for 101 attributes, no data is missing. Then we standardize the data matrix $\bar{A}$ so that each column of $\bar{A}$ has a zero mean and unit standard deviation. The dataset is from UCI machine learning repository, and it is available at the url: http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized.

Our goal is to predict the murder rate per 100,000 people as a linear function of the attributes, where at most k attributes can be nonzero. We include a bias term in our model, *i.e.*, in (SR) we set $A = [\bar{A} \mid \mathbf{1}]$. We use 10-fold cross-validation and vary k from 2 to 20.

Results. Figure 3 shows the RMS error for the training datasets, validation datasets, and the test dataset along with one-standard-error bands. The results for training, validation, and test datasets are reasonably similar for each value of k. This gives us confidence that the sparse regression model will have similar performance on new and unseen data. This also suggests that our model does not suffer from over-fitting. We also see that, for $k \geq 10$, none of the errors drop significantly. Scanning the RMS error on the test sets in Figure 3, we can expect that our prediction error on new data will be around 10.

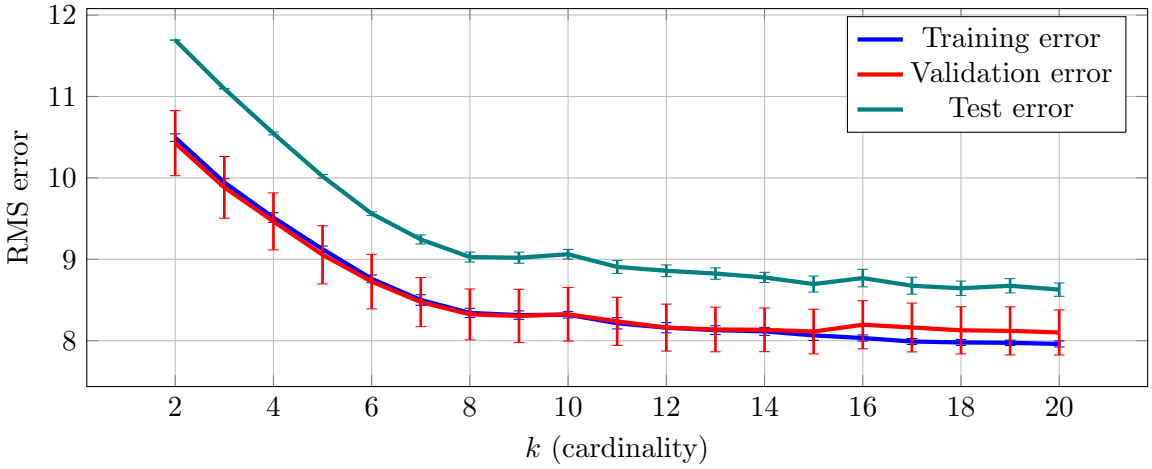


Figure 3: RMS error vs k (cardinality) for the murder rate detection problem.

3.2 Affine rank minimization problem

Problem description. In (SR), we set $\mathcal{X} := \{X \in \mathbf{R}^{m\times d} \mid \mathbf{rank}(X) \leq r, \|X\|_2 \leq \Gamma\}$, and $f(X) := \|\mathcal{A}(X) - b\|_2^2$. To compute the proximal operator of f, we use the formula in [20, §6.1.1]. Finally, we use the formula in [10, page 14] for projecting onto $\mathcal{X}$. Now we are in a position to apply the NExOS to this problem.

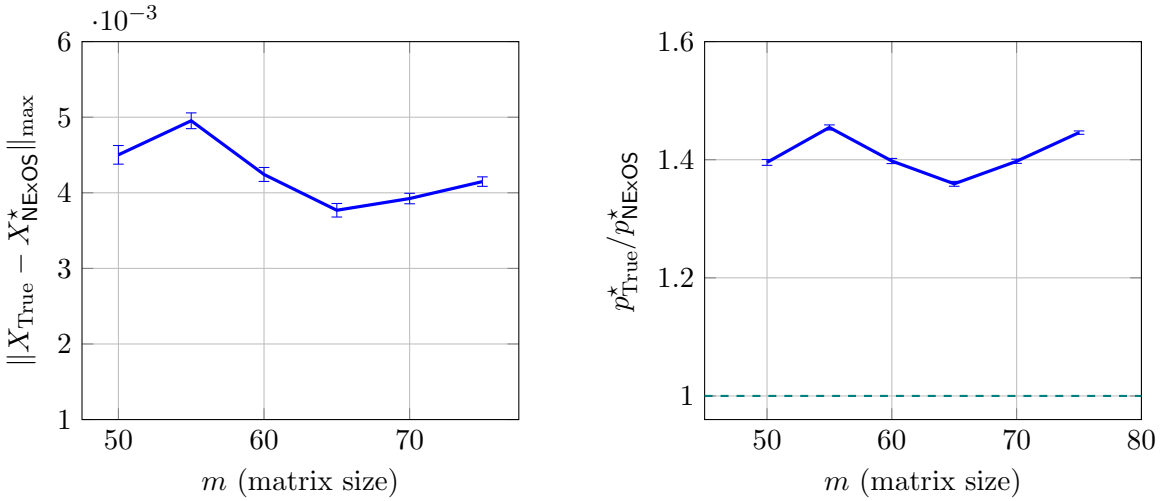
Summary of the experiments performed. *First,* we apply NExOS to solve (RM) for synthetic datasets, where we observe how the algorithm performs in recovering a low-rank matrix given noisy measurements. *Second,* we apply NExOS to a real-world dataset (`MovieLens 1M Dataset`) to see how our algorithm performs in solving a matrix-completion problem (which is a special case of (RM)).

3.2.1 Experiments and results for synthetic dataset

Data generation process and setup. We generate the data as follows, which is similar to [10]. We vary m (number of rows of the decision variable X) from 50 to 75 with a linear spacing of 5, where we take $d = 2m$, and rank to be equal to $m/10$ rounded to the nearest integer. For each value of m, we create 25 random instances as follows. The operator $\mathcal{A}$ is drawn from an iid normal distribution with $\mathcal{N}(0,1)$ entries. Similarly, we create the low rank matrix X_{True} with rank r, first drawn from an iid normal distribution with $\mathcal{N}(0,1)$ entries, and then truncating the singular values that exceed Γ to 0. Signal-to-noise ratio is taken to be around 20 by following the same method described for the sparse regression problem.

Results. The results displayed in the figures are averaged over 50 simulations for each value of m, and also show one-standard-error bands.

Figure 4a shows how well NExOS does in recovering the original matrix X_{True}. To quantify the recovery, we compute the max norm of the difference matrix $\|X_{\text{True}} - X^\star_{\mathsf{NExOS}}\|_{\max} = \max_{i,j} |X_{\text{True}}(i,j) - X^\star_{\mathsf{NExOS}}(i,j)|$, where the solution found by NExOS is denoted by $X^\star_{\mathsf{NExOS}}$. We see that the worst case component-wise error is very small in all the cases.



(a) Maximum absolute error in recovering the original matrix vs m

(b) Ratio of training losses of the true matrix X_{True} and the solution found by NExOS vs m

Figure 4: Affine rank minimization problem: comparison between solution found by NExOS and the true matrix

Finally, we show how the training loss of the solution $X^\star_{\text{NExOS}}$ computed by NExOS compares with the original matrix X_{True} in Figure 4b. Note that the ratio is larger than one in most cases, *i.e.*, NExOS finds a solution that has a smaller value compared to X_{True}. This is due to the fact that under the quite high signal-to-noise ratio the problem data can be explained better by another matrix with a lower training loss. That being said, $X^\star_{\text{NExOS}}$ is not too far from X_{True} component-wise as we saw in Figure 4a.

3.2.2 Experiments and results for real-world dataset: matrix completion problem

Description of the dataset. To investigate the performance of our problem on a real-world dataset, we consider the `MovieLens 1M Dataset`,which is available at the url: https://grouplens.org/datasets/movielens/1m/. This dataset contains 1,000,023 ratings for 3,706 unique movies (the dataset contains some repetitions in movie ratings and we have ignored them); these recommendations were made by 6,040 MovieLens users. The rating is on a scale of 1 to 5. If we construct a matrix of movie ratings by the users (also called the preference matrix), denoted by Z, then it is a matrix of 6,040 rows (each row corresponds to a user) and 3,706 columns (each column corresponds to a movie) with only 4.47% of the total entries are observed, while the rest of the entries are missing. We standardize the preference matrix Z (ignoring the missing entries) so that each column has a zero mean and unit standard deviation. Our goal is to complete this matrix, under the assumption that the matrix is low-rank. For more details about the model, we refer the reader to the discussion in [18, §8.1].

To gain confidence in the generalization ability of this model, we use an out-of-sample validation process. By random selection, we split the available data into a training set (80% of the total data) and a test set (20% of the total data). We use the training set as the input data for solving the underlying optimization process, and the held-out test set is used to compute the test error for each value of r, which is indicative of the generalization ability of the model. The best rank

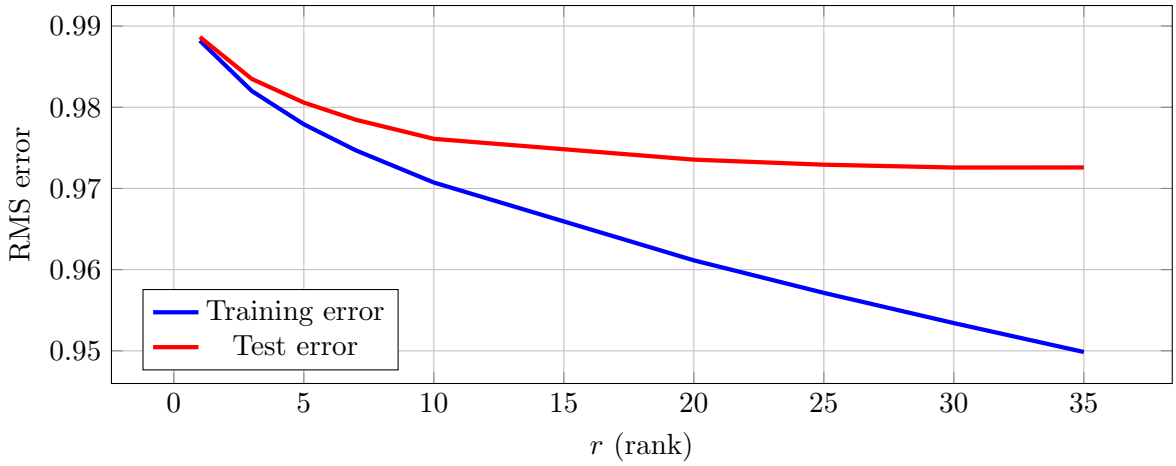


Figure 5: RMS error vs r (rank) for the matrix completion problem.

r corresponds to the point beyond which the improvement in the test error is rather minor. We tested rank values r ranging in $\{1, 3, 5, 7, 10, 20, 25, 30, 35\}$.

Matrix completion problem. The underlying optimization problem in this case is called a matrix completion problem. A matrix completion problem can be formulated as:

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in\Omega}(X_{ij} - Z_{ij})^2 + (\beta/2)\|X\|_F^2 \\ \text{subject to} & \mathbf{rank}(X) \leq r \\ & \|X\|_2 \leq \Gamma, \end{array} \tag{MC}$$

where $Z \in \mathbf{R}^{m\times d}$ is the matrix whose entries Z_{ij} are observable for $(i, j) \in \Omega$. Based on these observed entries, our goal is to construct a matrix $X \in \mathbf{R}^{m\times d}$ that has rank r. The problem above can be written as a special case of affine rank minimization problem (RM). Hence, computing projection onto the constraint set and the proximal operator is the same as in problem (RM). The upper bound Γ for $\|X\|_2$ is computed as follows. We construct a virtual matrix Y by filling each missing entries of standardized matrix Z with the absolute maximum of the available entries, which results in the maximum possible Frobenius norm of any preference matrix in our setup. Then we compute the Frobeinus norm $\|Y\|_F$ and set $\Gamma = \|Y\|_F$. As the spectral norm of a matrix is always less than its Frobenius norm, this gives a valid upper bound on $\|X\|_2$.

Results. Figure 5 shows the RMS error for the training datatest and test dataset for each value of rank r. The results for training and test datasets are reasonably similar for each value of r. We observe that beyond rank 15, the reduction in the test error is rather minor and going beyond this rank provides only diminishing return, which is a common occurrence for low-rank matrix approximation [48, §7.1]. Thus we can choose the optimal rank to be 15 for all practical purposes.

3.3 Factor analysis problem

Problem description. The factor analysis model with sparse noise (also known as low-rank factor analysis model) involves decomposing a given positive semidefinite matrix as a sum of a low-rank

positive semidefinite matrix and a diagonal matrix with nonnegative entries [27, page 191]. It can be posed as the following optimization problem [43]:

$$\begin{array}{ll} \text{minimize} & \|\Sigma - X - D\|_F^2 + (\beta/2)\left(\|X\|_F^2 + \|D\|_F^2\right) \\ \text{subject to} & D = \mathbf{diag}(d) \\ & d \geq 0 \\ & X \succeq 0 \\ & \mathbf{rank}(X) \leq r \\ & \Sigma - D \succeq 0 \\ & \|X\|_2 \leq \Gamma, \end{array} \tag{FA}$$

where $X \in \mathbf{S}^p$ and the diagonal matrix $D \in \mathbf{S}^p$ with nonnegative entries are the decision variables, and $\Sigma \in \mathbf{S}_+^p$, $r \in \mathbf{Z}_+$, and $\Gamma \in \mathbf{R}_{++}$ are the problem data.

A proper solution for (FA) requires that both X and D are positive semidefinite. The covariance matrix for the common parts of the variables, *i.e.*, $\Sigma - D$ has to be positive semidefinite, as otherwise statistical interpretations of the solution will be troublesome if not impossible [44, page 326].

In (FA), we set $\mathcal{X} := \{(X, D) \in \mathbf{S}^p \times \mathbf{S}^p \mid \|X\|_2 \leq \Gamma, \mathbf{rank}(X) \leq r, D = \mathbf{diag}(d), d \geq 0\}$, and $f(X, D) := \|\Sigma - X - D\|_F^2 + I_{\mathcal{P}}(X, D)$,where $I_{\mathcal{P}}$ denotes the indicator function of the convex set $\mathcal{P} = \{(X, D) \in \mathbf{S}^p \times \mathbf{S}^p \mid X \succeq 0, D = \mathbf{diag}(d), d \geq 0, d \in \mathbf{R}^p\}$. To compute the projection onto $\mathcal{X}$, we use the formula in [10, page 14] and the fact that $\mathbf{\Pi}_{\{y \mid y \geq 0\}}(x) = \max\{x, 0\}$, where pointwise max is used. The proximal operator for f at (X, D) can be computed by solving the following convex optimization problem:

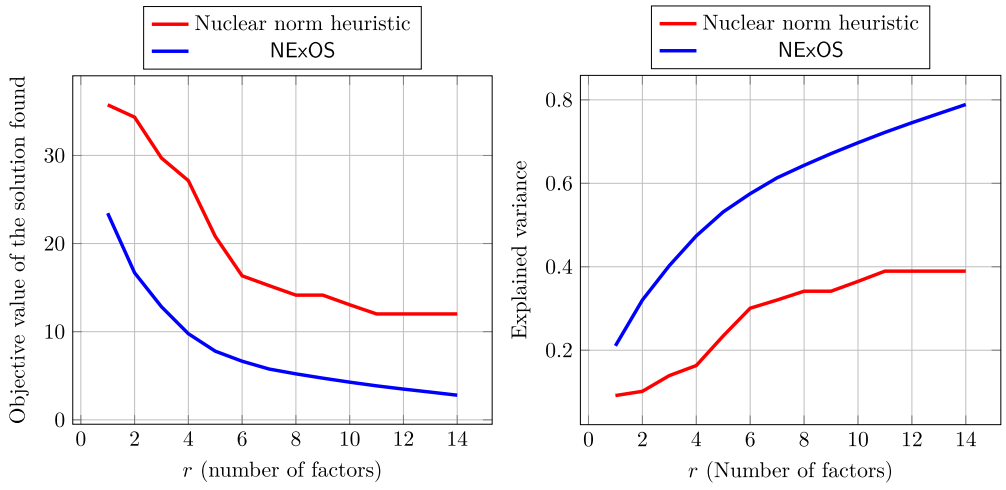
$$\begin{array}{ll} \text{minimize} & \left\|\Sigma - \widetilde{X} - \widetilde{D}\right\|_F^2 + (1/2\gamma)\|\widetilde{X} - X\|_F^2 + (1/2\gamma)\|\widetilde{D} - D\|_F^2 \\ \text{subject to} & \widetilde{X} \succeq 0,\ \widetilde{D} = \mathbf{diag}(\widetilde{d}), \\ & \Sigma - \widetilde{D} \succeq 0,\ \widetilde{d} \geq 0, \end{array}$$

where $\widetilde{X} \in \mathbf{S}_+^p$, and $\widetilde{d} \in \mathbf{R}_+^p$ (*i.e.*, $\widetilde{D} = \mathbf{diag}(\widetilde{d})$) are the optimization variables. Now we are in a position to apply NExOS to this problem.

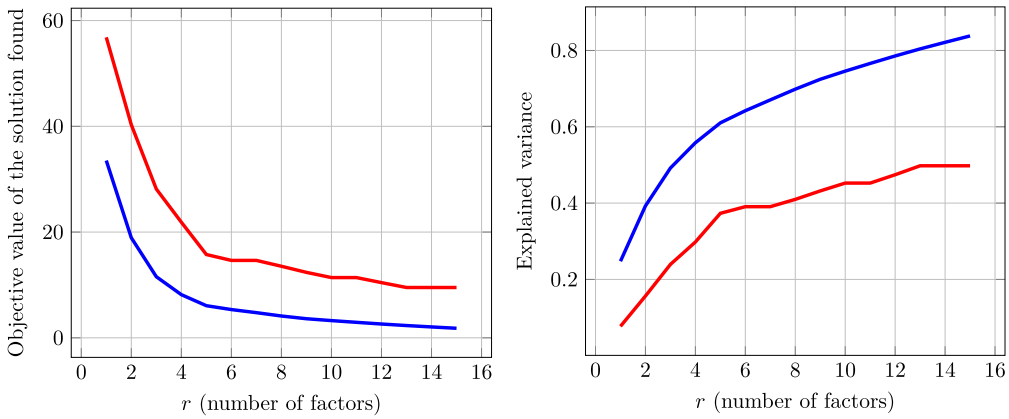
Comparison with nuclear norm heuristic. We compare the solution provided by NExOS to that of the nuclear norm heuristic, which is perhaps the most well-known heuristic to approximately solve (FA) [45]. This heuristic considers the following convex relaxation of (FA):

$$\begin{array}{ll} \text{minimize} & \|\Sigma - X - D\|_F^2 + \lambda \|X\|_* \\ \text{subject to} & D = \mathbf{diag}(d) \\ & d \geq 0,\ X \succeq 0 \\ & \Sigma - D \succeq 0,\ \|X\|_2 \leq \Gamma, \end{array} \tag{18}$$

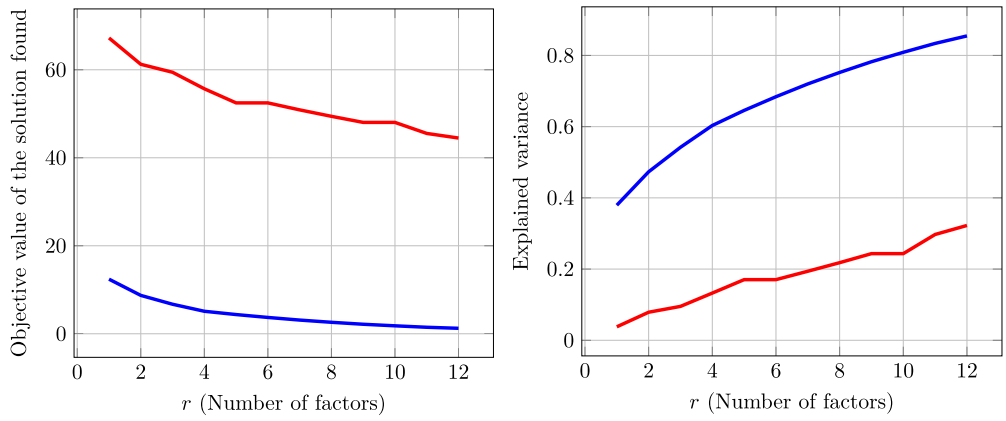
where λ is a positive parameter that is related to the rank of the decision variable X. Note that, as X is positive semidefinite, we have its nuclear norm $\|X\|_* = \mathbf{tr}(X)$; for this reason this heuristic is also called minimum trace factor analysis. The nuclear norm heuristic is similar to Lasso because it solves the relaxed problem (18) for different values of λ, and find the smallest value of λ for which solution to (18) satisfies $\mathbf{rank}(X) \leq r$.



(a) `bfi` dataset



(b) `neo` dataset



(c) `Harman74` dataset

Figure 6: Figure showing performance of NExOS in solving factor analysis problem for different datasets. Each row represents one dataset. The first and second column compares training loss and proportion of the variance explained of the solutions found by NExOS and the nuclear norm heuristic.

Performance measures. We consider two performance measures. First, we compare the training loss $\|\Sigma - X - D\|_F^2$ of the solutions found by NExOS and the nuclear norm heuristic. As both NExOS and the nuclear norm heuristic provide a point from the feasible set of (FA), such a comparison of training losses tells us which algorithm is providing a better quality solution.

Second, we compute the *proportion of explained variance*, which represents how well the r-common factors explain the residual covariance, *i.e.*, $\Sigma - D$. For a given r, input proportion of variance explained by the r common factors is given by: $\sum_{i=1}^{r} \sigma_i(X) / \sum_{i=1}^{p} \sigma_i(\Sigma - D)$, where X, D are inputs, that correspond to solutions found by NExOS or the nuclear norm heuristic. As r increases, the explained variance increases to 1. The higher the value of the explained variance for a certain solution, the better is the quality of the solution from a statistical point of view.

Description of the datasets. We consider three different real-world bench-mark datasets that are popularly used for factor analysis. These datasets can be found in the `R` libraries `datasets`, and `psych`. The `bfi` dataset contains 2800 observations with 28 variables (25 personality self-reported items and 3 demographic variables) and is available at: https://www.rdocumentation.org/packages/psych/versions/1.0-93/topics/bfi. The `neo` dataset has 1000 measurements for 30 variables and is available at: https://www.rdocumentation.org/packages/psych/versions/1.8.12/topics/neo. The `Harman74` dataset has 145 observations on 24 variables and is available at: https://rdrr.io/r/datasets/Harman74.cor.html.

Setup. In applying NExOS for the factor analysis problem, we initialize our iterates with $Z_0 := \Sigma$ and $z_0 := \mathbf{0}$. All the other parameters are kept at their default values as stated in the beginning of §3. For each dataset, we vary the number of factors from 1 to $\lfloor p/2 \rfloor$, where p is the size of the underlying matrix Σ.

Results. Figure 6 shows performance of NExOS in solving the factor analysis problem for different datasets, with each row representing one dataset.

The first column compares the training loss of the solution found by NExOS and the nuclear norm heuristic. We see that for all the datasets, NExOS finds a solution with a training loss that is considerably smaller than that of the nuclear norm heuristic.

The second column shows the proportion of variance explained by the algorithms considered for the datasets considered (higher is better). We see that in terms of the proportion of explained variance, NExOS delivers larger values than that of the nuclear norm heuristic for different values of r, which is indeed desirable. NExOS consistently provides solutions with better objective value and explained variance compared to the nuclear norm heuristic.

4 Conclusion

In this paper, we have presented NExOS, a novel first-order algorithm to solve optimization problems with convex cost functions over nonconvex constraint sets— a problem structure that is satisfied by a wide range of nonconvex machine learning problems including sparse and low-rank optimization. We have shown that, under mild technical conditions, NExOS is able to find a locally optimal point of the original problem by solving a sequence of penalized problems with strictly decreasing

penalty parameters. We have implemented our algorithm in the `Julia` package `NExOS.jl` and have extensively tested its performance on a wide variety of nonconvex learning problems. We have demonstrated that NExOS is able to compute high quality solutions at a speed that is competitive with tailored algorithms.

References

[1] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "glmnet: Lasso and elastic-net regularized generalized linear models," *R package version*, vol. 1, no. 4, pp. 1–24, 2009.

[2] J. D. Blanchard, J. Tanner, and K. Wei, "CGIHT: conjugate gradient iterative hard thresholding for compressed sensing and matrix completion," *Information and Inference: A Journal of the IMA*, vol. 4, no. 4, pp. 289–327, 2015.

[3] J. D. Blanchard and J. Tanner, "Performance comparisons of greedy algorithms in compressed sensing," *Numerical Linear Algebra with Applications*, vol. 22, no. 2, pp. 254–282, 2015.

[4] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[5] ——, "Iterative thresholding for sparse approximations," *Journal of Fourier analysis and Applications*, vol. 14, no. 5-6, pp. 629–654, 2008.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[7] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," *International Journal of Control*, pp. 1–11, 2017.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[9] R. Takapoui, "The alternating direction method of multipliers for mixed-integer optimization applications," Ph.D. dissertation, Stanford University, 2017.

[10] S. Diamond, R. Takapoui, and S. Boyd, "A general system for heuristic minimization of convex functions over non-convex sets," *Optimization Methods and Software*, vol. 33, no. 1, pp. 165–193, 2018.

[11] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2017, vol. 408.

[12] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[13] B. T. Polyak, "Introduction to optimization. Translations series in mathematics and engineering," *Optimization Software*, 1987.

[14] A. Beck, *First-Order Methods in Optimization*. SIAM, 2017, vol. 25.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[16] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[17] P. Giselsson and S. Boyd, "Linear convergence and metric selection for Douglas-Rachford splitting and ADMM," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 532–544, Feb 2017.

[18] P. Jain and P. Kar, "Non-convex optimization for machine learning," *Foundations and Trends® in Machine Learning*, vol. 10, no. 3-4, pp. 142–336, 2017.

[19] R. Poliquin, R. T. Rockafellar, and L. Thibault, "Local differentiability of distance functions," *Transactions of the American Mathematical Society*, vol. 352, no. 11, pp. 5231–5249, 2000.

[20] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[21] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[22] F. H. Clarke, R. J. Stern, and P. R. Wolenski, "Proximal smoothness and the lower-$\mathcal{C}^2$ property," *Journal of Convex Analysis*, vol. 2, no. 1-2, pp. 117–144, 1995.

[23] J.-P. Vial, "Strong and weak convexity of sets and functions," *Mathematics of Operations Research*, vol. 8, no. 2, pp. 231–259, 1983.

[24] A. Shapiro, "Existence and differentiability of metric projections in Hilbert spaces," *SIAM Journal on Optimization*, vol. 4, no. 1, pp. 130–141, 1994.

[25] D. R. Luke, "Prox-regularity of rank constraint sets and implications for algorithms," *Journal of Mathematical Imaging and Vision*, vol. 47, no. 3, pp. 231–238, Nov 2013.

[26] W. Rudin, *Principles of Mathematical Analysis*. McGraw-hill New York, 1986.

[27] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: The lasso and generalizations*. Taylor & Francis, 2015.

[28] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, 2006.

[29] E. Candès, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, 2008.

[30] D. Bertsimas and B. Van Parys, "Sparse hierarchical regression with polynomials," *Machine Learning*, 2020.

[31] D. Bertsimas, A. King, and R. Mazumder, "Best subset selection via a modern optimization lens," *Annals of Statistics*, 2016.

[32] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of Machine Learning Research*, 2010.

[33] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, 2009.

[34] A. Gress and I. Davidson, "A flexible framework for projecting heterogeneous data," in *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, 2014.

[35] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013.

[36] V. Srikumar and C. D. Manning, "Learning distributed representations for structured output prediction," in *Advances in Neural Information Processing Systems*, 2014.

[37] F. Bach, "Sharp analysis of low-rank kernel matrix approximations," in *Journal of Machine Learning Research*, 2013.

[38] M. Hardt, R. Meka, P. Raghavendra, and B. Weitz, "Computational limits for Matrix Completion," in *Journal of Machine Learning Research*, 2014.

[39] M. Fazel, E. Candès, B. Recht, and P. Parrilo, "Compressed sensing and robust recovery of low rank matrices," in *Conference Record - Asilomar Conference on Signals, Systems and Computers*, 2008.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[41] A. L. Dontchev and R. T. Rockafellar, *Implicit functions and solution mappings*. Springer, 2009, vol. 543.

[42] F. Bernard, L. Thibault, and N. Zlateva, "Prox-regular sets and epigraphs in uniformly convex Banach spaces: various regularities and other properties," *Transactions of the American Mathematical Society*, vol. 363, no. 4, pp. 2211–2247, 2011.

[43] D. Bertsimas, M. S. Copenhaver, and R. Mazumder, "Certifiably optimal low rank factor analysis," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 907–959, 2017.

[44] J. M. ten Berge, "Some recent developments in factor analysis and the search for proper communalities," in *Advances in data science and classification*. Springer, 1998, pp. 325–334.

[45] J. Saunderson, V. Chandrasekaran, P. Parrilo, and A. S. Willsky, "Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 4, pp. 1395–1416, 2012.

[46] A. V. Fiacco and G. P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM, 1990.

[47] L. Stella, N. Antonello, and M. Falt, "ProximalOperators. jl," 2020.

[48] J. Lee, S. Kim, G. Lebanon, Y. Singer, and S. Bengio, "LLORMA: Local low-rank matrix approximation," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 442–465, 2016.

[49] A. Auslender, "Stability in mathematical programming with nondifferentiable data," *SIAM Journal on Control and Optimization*, vol. 22, no. 2, pp. 239–254, 1984.

[50] R. T. Rockafellar, "Characterizing firm nonexpansiveness of prox mappings both locally and globally," *Journal of Nonlinear and convex Analysis*, vol. 22, no. 5, 2021.

[51] D. Bertsimas, A. King, and R. Mazumder, "Best subset selection via a modern optimization lens," *The annals of statistics*, pp. 813–852, 2016.

[52] D. Bertsimas and J. Dunn, *Machine Learning Under a Modern Optimization Lens*. Dynamic Ideas, MA, 2019.

[53] H. Hazimeh and R. Mazumder, "Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms," *Operations Research*, vol. 68, no. 5, pp. 1517–1537, 2020.

[54] D. Bertsimas, B. Van Parys *et al.*, "Sparse high-dimensional regression: Exact scalable algorithms and phase transitions," *The Annals of Statistics*, vol. 48, no. 1, pp. 300–323, 2020.

[55] D. Bertsimas, R. Cory-Wright, and J. Pauphilet, "Mixed-projection conic optimization: A new paradigm for modeling rank constraints," *Operations Research (accepted)*, 2020.

[56] R. Correa, A. Jofre, and L. Thibault, "Characterization of lower semicontinuous convex functions," *Proceedings of the American Mathematical Society*, pp. 67–72, 1992.

[57] A. M. Tillmann, D. Bienstock, A. Lodi, and A. Schwartz, "Cardinality minimization, constraints, and regularization: A survey," *arXiv preprint arXiv:2106.09606*, 2021.

[58] R. Poliquin and R. Rockafellar, "Prox-regular functions in variational analysis," *Transactions of the American Mathematical Society*, vol. 348, no. 5, pp. 1805–1838, 1996.

[59] M. Udell, C. Horn, R. Zadeh, and S. Boyd, "Generalized low rank models," *Foundations and Trends in Machine Learning*, vol. 9, no. 1, 2016. [Online]. Available: http://dx.doi.org/10.1561/2200000055

[60] K.-S. Jun, R. Willett, S. Wright, and R. Nowak, "Bilinear bandits with low-rank structure," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3163–3172.

A Proof and derivation to results in §1

A.1 Lemma regarding prox-regularity of intersection of sets

Lemma 3. *Consider the nonempty constraint set $\mathcal{X} = \mathcal{C} \bigcap \mathcal{N} \subseteq \mathbf{E}$, where $\mathcal{C}$ is compact and convex, and $\mathcal{N}$ is prox-regular at $x \in \mathcal{X}$. Then $\mathcal{X}$ is prox-regular at x.*

Proof to Lemma 3. To prove this result we record the following result from [42], where by $d_{\mathcal{S}}(x)$ we denote the Euclidean distance of a point x from the set $\mathcal{S}$, and $\overline{\mathcal{S}}$ denotes closure of a set $\mathcal{X}$.

Lemma 4 (Intersection of prox-regular sets [42, Corollary 7.3(a)])**.** *Let $\mathcal{S}_1, \mathcal{S}_2$ be two closed sets in $\mathbf{E}$, such that $\mathcal{S} = \mathcal{S}_1 \bigcap \mathcal{S}_2 \neq \emptyset$ and both $\mathcal{S}_1, \mathcal{S}_2$ are prox-regular at $x \in \mathcal{S}$. If $\mathcal{S}$ is metrically calm at x, i.e., if there exist some $\varsigma > 0$ and some neighborhood of x denoted by $\mathcal{B}$ such that*

$$d_{\mathcal{S}}(y) \leq \varsigma \left(d_{\mathcal{S}_1}(y) + d_{\mathcal{S}_2}(y) \right) \tag{19}$$

for all $y \in \mathcal{B}$, then $\mathcal{S}$ is prox-regular at x.

Now we start the proof to Lemma 3.

Proof. (proof to Lemma 3) By definition, projection onto $\mathcal{N}$ is single-valued on some open ball $B(x; a)$ with center x and radius $a > 0$ [19, Theorem 1.3]. The set $\mathcal{C}$ is compact and convex, hence projection onto $\mathcal{C}$ is single-valued around every point, hence single-valued on $B(x; a)$ as well [11, Theorem 3.14, Remark 3.15]. Note that for any $y \in B(x; a)$, $d_{\mathcal{X}}(y) = 0$ if and only if both $d_{\mathcal{C}}(y)$ and $d_{\mathcal{N}}(y)$ are zero. Hence, for any $y \in B(x; a) \bigcap \mathcal{X}$, the metrically calmness condition (19) is trivially satisfied. Next, recalling that the distance from a closed set is continuous [12, Example 9.6], over the compact set $\overline{B(x; a) \setminus \mathcal{X}}$, define the function

$$h(y) = \begin{cases} \frac{d_{\mathcal{X}}(y)}{d_{\mathcal{C}}(y) + d_{\mathcal{N}}(y)}, & \text{if } y \notin \mathcal{X} \\ 1, & \text{else,} \end{cases}$$

which is upper-semicontinuous over $\overline{B(x; a) \setminus \mathcal{X}}$, hence it will attain a maximum $\varsigma > 0$ over $\overline{B(x; a) \setminus \mathcal{X}}$ [26, Theorem 4.16], thus satisfying the metrically calmness condition (19) on $B(x; a) \setminus \mathcal{X}$ as well. Hence, using Lemma 4, the constraint set $\mathcal{X}$ is prox-regular at x. □

A.2 Constructing the inner algorithm of NExOS from Douglas-Rachford splitting.

We now discuss how to construct Algorithm 2 by applying Douglas-Rachford splitting to ($\mathcal{P}_\mu$). If we apply Douglas-Rachford splitting [11, page 401] to ($\mathcal{P}_\mu$) with penalty parameter μ, we have the following variant with three sub-iterations:

$$\begin{aligned} x^{n+1} &= \mathbf{prox}_{\gamma f}\left(z^n\right) \\ y^{n+1} &= \mathbf{prox}_{\gamma \mu_{\mathbb{q}}}\left(2x^{n+1} - z^n\right) \\ z^{n+1} &= z^n + y^{n+1} - x^{n+1}, \end{aligned} \tag{DRS}$$

where the proximal operator is defined in (4). In (DRS), x^n, y^n and z^n are the iterates, the superscript $n \in \mathbf{N}$ denotes the iteration counter, and the subscript μ indicates the current value of

the penalty parameter. The first sub-iterate $\mathbf{prox}_{\gamma f}(z^n)$ is always single-valued and computing it is equivalent to solving a convex optimization problem, which can be done in closed form for many relevant cost functions in machine learning [14, pp. 449-450]. With a slight abuse of notation, by $\mathbf{prox}_{\gamma\, \mu_{\mathbb{Q}}}\left(2x^{n+1}-z^n\right)$ we denote one arbitrary solution in (DRS) rather than the entire set-valued map as in the original definition (4). This does not cause a problem technically, because our algorithm is guaranteed to operate at points where the above-mentioned map is single-valued by exploiting the prox-regularity of the constraint set. The computational cost for $\mathbf{prox}_{\gamma\, \mu_{\mathbb{Q}}}$ is the same as computing a projection onto the constraint set $\mathcal{X}$, as we will show in Lemma 5 below.

Lemma 5 (Computing $\mathbf{prox}_{\gamma\, \mu_{\mathbb{Q}}}(x)$)**.** *Consider the nonconvex compact constraint set $\mathcal{X}$ in ($\mathcal{P}$). Denote $\kappa = 1/(\beta\gamma+1) \in [0,1]$ and $\theta = \mu/(\gamma\kappa+\mu) \in [0,1]$. Then, for any $x \in \mathbf{E}$, and for any $\mu, \beta, \gamma > 0$, we have*

$$\mathbf{prox}_{\gamma\, \mu_{\mathbb{Q}}}(x) = \theta\kappa x + (1-\theta)\, \mathbf{\Pi}\left(\kappa x\right).$$

Proof. Proof follows from [14, Theorem 6.13, Theorem 6.63]. It should be noted that [14, Theorem 6.13, Theorem 6.63] assume convexity of the functions in the theorem statements, but its proof does not require convexity and works for nonconvex functions as well. □

Combining (DRS), [14, Theorem 6.13], and Lemma 5, we arrive at Algorithm 2.

B Proofs and derivations to the results in §2

B.1 Modifying NExOS for nonsmooth and convex cost function

We now discuss how to modify NExOS when the cost function is nonsmooth and convex. The optimization problem in this case, where the positive regularization parameter is denoted by $\widetilde{\beta}$, is given by:

$$\text{minimize} \quad \phi(x) + (\widetilde{\beta}/2)\|x\|^2 + \iota(x) \tag{20}$$

where the setup is same as ($\mathcal{P}$), except the cost function $\phi : \mathbf{E} \to \mathbf{R} \cup \{+\infty\}$ is lower-semicontinuous, proper, and convex.

For this cost function ϕ, its Moreau envelope ${}^{\nu}\phi$, for any $\nu > 0$, has the following desirable features:

1. For every $x \in \mathbf{E}$ we have ${}^{\nu}\phi(x) \leq \phi(x)$ and ${}^{\nu}\phi(x) \to \phi(x)$ as $\nu \to 0$ [12, Theorem 1.25].
2. We have $x^\star \in \operatorname{argmin}_{x\in\mathbf{E}} \phi(x)$ if and only if $x^\star \in \operatorname{argmin}_{x\in\mathbf{E}} \phi(x)\, {}^{\nu}\phi(x)$ with the minimizer $x^\star$ satisfying $\phi(x^\star) = {}^{\nu}\phi(x^\star)$ [11, Corollary 17.5].
3. The Moreau envelope ${}^{\nu}\phi$ is convex, and smooth (*i.e.*, it is differentiable and its gradient is Lipschitz continuous) everywhere irrespective of the differentiability or smoothness of the original function ϕ. The gradient is given by: ${}^{\nu}\phi(x) = \left(x - \mathbf{prox}_{\nu\phi}(x)\right)/\nu$, which is $(1/\nu)-$Lipschitz continuous [11, Proposition 12.29].

The properties above, make ${}^{\nu}\phi$ a smooth approximation of ϕ for a small enough ν. Let $\beta := \widetilde{\beta}/2$. For a ν that is arbitrarily small, define the following β strongly convex and $(\nu^{-1}+\beta)$-smooth function:

$$f := {}^{\nu}\phi(\cdot) + (\beta/2)\|\cdot\|^2. \tag{21}$$

which makes the following optimization problem an arbitrarily close approximation to (20):

$$\text{minimize} \quad f + (\beta/2)\|x\|^2 + \iota(x) \tag{22}$$

The setup in (22) is now the same as ($\mathcal{P}$). The cost to compute the proximal operator of f in (21) is the same as computing the proximal operator of ϕ and is given by

$$\mathbf{prox}_{\gamma f}(x) = \frac{1}{\gamma\beta+1}x + \frac{\gamma(\gamma\beta+1)^{-1}}{\gamma(\gamma\beta+1)^{-1}+\nu}\left(\mathbf{prox}_{(\gamma(\gamma\beta+1)^{-1}+\nu)\phi}\left(\frac{1}{\gamma\beta+1}x\right) - \frac{1}{\gamma\beta+1}x\right),$$

which follows from [14, Theorem 6.13, Theorem 6.63]. Then, we apply NExOS to (22) and proceed in the same manner as discussed earlier.

B.2 Proof to Proposition 1

B.2.1 Proof to Proposition 1(i)

We prove (i) in three steps. In the *first step*, we show that for any $\mu > 0$, $f +^{\mu}\!\iota$ will be differentiable on some $B(\bar{x}; r_{\text{diff}})$ with $r_{\text{diff}} > 0$. In the *second step*, we then show that, for any $\mu \in (0, 1/\beta]$, $f +^{\mu}\!\iota$ will be strongly convex and differentiable on some $B(\bar{x}; r_{\text{cvxdiff}})$. In the *third step*, we will show that there exist $\mu_{\max} > 0$ such that for any $\mu \in (0, \mu_{\max}]$, $f +^{\mu}\!\iota$ will be strongly convex and smooth on some $B(\bar{x}; r_{\max})$ and will attain the unique local minimum x_μ in this ball.

Proof of the first step. To prove the first step, we start with the following lemma regarding differentiability of ${}^{\mu}\iota$.

Lemma 6 (Differentiability of ${}^{\mu}\iota$)**.** *Let $\bar{x}$ be a local minimum to ($\mathcal{P}$), where Assumptions 1 and 2 hold. Then there exists some $r_{\text{diff}} > 0$ such that for any $\mu > 0$,*

(i) the function ${}^{\mu}\iota$ is differentiable on $B(\bar{x}; r_{\text{diff}})$ with derivative $\nabla\, {}^{\mu}\iota = (1/\mu)(\mathbb{I} - \mathbf{\Pi})$, and

(ii) the projection operator $\mathbf{\Pi}$ onto $\mathcal{X}$ is single-valued and Lipschitz continuous on $B(\bar{x}; r_{\text{diff}})$.

Proof. From [19, Theorem 1.3(e)], there exists some $r_{\text{diff}} > 0$ such that the function d^2 is differentiable on $B(\bar{x}; r_{\text{diff}})$. As ${}^{\mu}\iota = (1/2\mu)d^2$ from (3), it follows that for any $\mu > 0$, ${}^{\mu}\iota$ is differentiable on $B(\bar{x}; r_{\text{diff}})$ which proves the first part of (i). The second part of (i) follows from the fact that $\nabla d^2(x) = 2\,(x - \mathbf{\Pi}(x))$ whenever d^2 is differentiable at x [19, page 5240]. Finally, from [19, Lemma 3.2], whenever d^2 is differentiable at a point, projection $\mathbf{\Pi}$ is single-valued and Lipschitz continuous around that point, and this proves (ii). □

Due to the lemma above, $f +^{\mu}\!\iota$ will be differentiable on $B(\bar{x}; r_{\text{diff}})$ with $r_{\text{diff}} > 0$, as f and $(\beta/2)\|\cdot\|^2$ are differentiable. Also, due to Lemma 6(ii), projection operator $\mathbf{\Pi}$ is $\widetilde{L}$-Lipschitz continuous on $B(\bar{x}; r_{\text{diff}})$. This proves the first step.

Proof of the second step. To prove this step, we are going to record (1) the notion of general subdifferential of a function, followed by (2) the definition of prox-regularity of a function and its connection with prox-regular set, and (3) a helper lemma regarding convexity of the Moreau envelope under prox-regularity.

Definition 3 (Fenchel, Fréchet, and general subdifferential)**.** For any lower-semicontinuous function $h : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$, its Fenchel subdifferential ∂h is defined as [56, page 1]:

$$u \in \partial h(x) \Leftrightarrow (\forall y \in \mathbf{R}^n) \quad h(y) \geq h(x) + \langle u \mid y - x \rangle .$$

For the function h, its Fréchet subdifferential $\partial^F h$ (also known as regular subdifferential) at a point x is defined as [56, Definition 2.5]:

$$u \in \partial^F h(x) \Leftrightarrow \liminf_{y \to 0} \frac{h(x+y) - h(x) - \langle u \mid y \rangle}{\|y\|} \geq 0.$$

Finally, the general subdifferential of h, denoted by$\partial^G h$, is defined as [50, Equation (2.8)]:

$$u \in \partial^G h(x) \Leftrightarrow \left(\exists (x_n, u_n) \in \mathbf{gra}\, \partial^F h\right) \quad u_n \to u, x_n \to x, f(x_n) \to f(x).$$

If h is additionally convex, then $\partial h = \partial^F h = \partial^G h$ [56, Property (2.3), Property 2.6].

Definition 4 (Connection between prox-regularity of a function and a set [58, Definition 1.1])**.** A function $h : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ that is finite at $\tilde{x}$ is prox-regular at $\tilde{x}$ for $\tilde{\nu}$, where $\tilde{\nu} \in \partial^G h(\tilde{x})$, if h is locally l.s.c. at $\tilde{x}$ and there exist a distance $\sigma > 0$ and a parameter $\rho > 0$ such that whenever $\|x' - \tilde{x}\| < \sigma$ and $\|x - \tilde{x}\| < \sigma$ with $x' \neq x$, $\|h(x) - h(\tilde{x})\| < \sigma$, $\|\nu - \tilde{\nu}\| < \sigma$ with $\nu \in \partial^G h(x)$, we have

$$h(x') > h(x) + \left\langle \nu \mid x' - x \right\rangle - \frac{\rho}{2}\|x' - x\|^2.$$

A set $\mathcal{S}$ is prox-regular at $\tilde{x}$ for $\tilde{\nu}$ if we have the indicator function $\iota_{\mathcal{S}}$ is prox-regular at $\tilde{x}$ for $\tilde{\nu} \in \partial^G \iota_{\mathcal{S}}(\tilde{x})$ [58, Proposition 2.11]. The set $\mathcal{S}$ is prox-regular at $\tilde{x}$ if it is prox-regular at $\tilde{x}$ for all $\tilde{\nu} \in \partial^G \iota_{\mathcal{S}}(\tilde{x})$ [12, page 612].

We have the following helper lemma from [58]. For completeness, we provide the proof here.

Lemma 7 ([58, Theorem 5.2])**.** *Consider a function h which is lower semicontinuous at 0 with $h(0) = 0$ and there exists $\rho > 0$ such that $h(x) > -\frac{\rho}{2}\|x\|^2$ for any $x \neq 0$. Let h be prox-regular at $\tilde{x} = 0$ and $\tilde{\nu} = 0$ with respect to σ and ρ, and let $\lambda \in (0, 1/\rho)$. Then, on some neighborhood of 0*

$$^{\lambda}h + \frac{\rho}{2(1 - \lambda\rho)}\|\cdot\|^2 \tag{23}$$

is convex, where $^{\lambda}h$ is the Moreau envelope of h with parameter λ.

Proof. In this proof, we use the notion of a monotone operator. Recall that, a set-valued operator $\mathbb{A} : \mathbf{E} \rightrightarrows \mathbf{E}$ is monotone on $X \times U$, if for any $(x, u), (y, v) \in X \times U$ with $u \in \mathbb{A}(x), v \in \mathbb{A}(y)$, we have $\langle u - v \mid x - y \rangle \geq 0$. Let $\lambda \in (0, 1/\rho)$. The proof the first statement follows from the proof to [58, Theorem 5.2], but we present it here for completeness. Define the operator $\mathbb{G}$ satisfying

$$\mathbb{G}(x) = \begin{cases} \{\nu \in \partial^G h(x) \mid \|\nu - \tilde{\nu}\| < \sigma\}, & \text{if } \|x - \tilde{x}\| < \sigma, |h(x) - h(\tilde{x})| < \sigma, \\ \emptyset, & \text{else,} \end{cases}$$

which is also known as the h-attentive σ-localization of $\partial^G h$ around $(\tilde{x}, \tilde{\nu})$ [58, Definition 3.1]. Because h is prox-regular at $\tilde{x} = 0$ and $\tilde{\nu} = 0$ with respect to σ and ρ, the operator $\mathbb{G} + \rho\mathbb{I}$ will

be monotone on $B(\tilde{x};\sigma)\times B(\tilde{\nu};\sigma)$ due to [58, Remark 3.3] . Under this setup, ${}^{\lambda}h$ is differentiable on some $B(\tilde{x};r_{\text{diff}})$ due to [12, Theorem 4.4]. This implies that the operator $\nabla({}^{\lambda}h)+(\rho/1-\lambda\rho)\mathbb{I}$ will be monotone and single-valued on $B(\tilde{x};\min\{\sigma,r_{\text{diff}}\})\times B(\tilde{\nu};\sigma)$, which follows from [58, Lemma 4.5, and Proposition 4.6]. But, on $B(\tilde{x};\min\{\sigma,r_{\text{diff}}\})$, the function in (23) will be differentiable and locally Lipschitz from [58, Proposition 4.2], respectively, and $\nabla({}^{\lambda}h)+(\rho/1-\lambda\rho)\mathbb{I}$, which is the gradient mapping of the aforementioned function on $B(\tilde{x};\min\{\sigma,r_{\text{diff}}\})\times B(\tilde{\nu};\sigma)$, is monotone. Because for a differentiable function, Fréchet subdifferential is equal to the gradient mapping [12, Exercise 8.8(a)] and monotonicity of the Fréchet subdifferential for a locally Lipschitz function imply its convexity [56, Remark after Property 2.7, Theorem 3.8], the function in (23) is convex on $B(\tilde{x};\min\{\sigma,r_{\text{diff}}\})$. □

Now we start proving step 2 earnestly. To prove this result, we assume $\bar{x}=0$. This does not cause any loss of generality because this is equivalent to transferring the coordinate origin to the optimal solution and prox-regularity of a set and strong convexity of a function is invariant under such a coordinate transformation. .

First, note that the indicator function of our constraint closed set $\mathcal{X}$ is lower semicontinuous due to [12, Remark after Theorem 1.6, page 11], and as $\bar{x}$, the local minimizer lies in $\mathcal{X}$, we have $\iota(\bar{x})=0$. The set $\mathcal{X}$ is prox-regular at $\bar{x}$ for all $\nu\in\partial^{G}\iota(x)$ per our setup, so using Definition 4, we have ι prox-regular at $\bar{x}=0$ for $\bar{\nu}=0\in\partial^{G}\iota(\bar{x})$ (because $\bar{x}\in\mathcal{X}$, we will have 0 as a subgradient of the $\partial\iota(\bar{x})$) with respect to some distance $\sigma>0$ and parameter $\rho>0$.

Note that the indicator function satisfies $\iota(x)=c\iota(x)$ for any $c>0$ due to its definition, so $u\in\partial^{G}\iota(x)\Leftrightarrow cu\in c\partial^{G}\iota(x)=\partial(c\iota^{G}(x))=\partial\iota^{G}(x)$ [12, Equation 10(6)] In our setup, we have $\mathcal{X}$ prox-regular at $\bar{x}$. So, setting $h:=\iota,\tilde{x}:=\bar{x}=0$, $\tilde{\nu}:=\bar{\nu}=0$, and $\nu:=u/(\beta/2\rho)$ in Definition 4, we have ι is also prox-regualr at $\bar{x}=0$ for $\bar{\nu}=0$ with respect to distance $\sigma\min\{1,\beta/2\rho\}$ and parameter $\beta/2$.

Next, because the range of the indicator function is $\{0,\infty\}$, we have

$$\iota(x)>-\frac{\rho}{2}\|x\|^2$$

for any $x\neq 0$. So, we have all the conditions of Theorem 7 satisfied. Hence, applying Theorem 7, we have for any $\mu\in(0,2/\beta)$

$$\begin{aligned}{}^{\mu}\iota+\frac{\rho}{2(1-\mu\rho)}\|\cdot\|^2&=\frac{1}{2\mu}d^2+\frac{\rho}{2(1-\mu\rho)}\|\cdot\|^2\\&=\frac{1}{2\mu}\left(d^2+\frac{\rho\mu}{1-\mu\rho}\|\cdot\|^2\right)\\&=\frac{1}{2\mu}\left(d^2+\frac{(\beta/2)\mu}{1-\mu(\beta/2)}\|\cdot\|^2\right)\end{aligned}$$

convex and differentiable on $B\left(\bar{x};\min\left\{\sigma\min\{1,\beta/2\rho\},r_{\text{diff}}\right\}\right)$, where r_{diff} comes from Lemma 6. As r_{diff} in this setup does not depend on μ, the ball $B\left(\bar{x};\min\left\{\sigma\min\{1,\beta/2\rho\},r_{\text{diff}}\right\}\right)$ does not depend on μ either. Finally, note that in our exterior-point minimization function we have

$${}^{\mu}\mathbb{1}={}^{\mu}\iota+(\beta/2)\|\cdot\|^2=\frac{1}{2\mu}d^2+\frac{\beta}{2}\|\cdot\|^2=\frac{1}{2\mu}\left(d^2+\beta\mu\|\cdot\|^2\right).$$

So if we take $\mu \leq \frac{1}{\beta}$, then we have

$$\frac{(\beta/2)\mu}{1-\mu(\beta/2)} \leq \beta\mu,$$

and on the ball $B\left(\bar{x}; \min\left\{\sigma \min\{1, \beta/2\rho\}, r_{\text{diff}}\right\}\right)$, the function ${}^{\mu}\!\!\downarrow$ will be convex and differentiable. But f is strongly-convex and smooth, so $f + {}^{\mu}\!\!\downarrow$ will be strongly convex and differentiable on $B\left(\bar{x}; \min\left\{\sigma \min\{1, \beta/2\rho\}, r_{\text{diff}}\right\}\right)$ for $\mu \in (0, 1/\beta]$. This proves step 2.

Proof of the third step. As point $\bar{x} \in \mathcal{X}$ is a local minimum of ($\mathcal{P}$), from Definition 2, there is some $r > 0$ such that for all $y \in \overline{B}(\bar{x}; r)$, we have

$$f(\bar{x}) + \frac{\beta}{2}\|\bar{x}\|^2 < f(y) + \frac{\beta}{2}\|y\|^2 + \iota(y).$$

Then, due to the first two steps, for any $\mu \in (0, 1/\beta]$, the function $f + {}^{\mu}\!\!\downarrow$ will be strongly convex and differentiable on $B\left(\bar{x}; \min\left\{\sigma \min\{1, \beta/2\rho\}, r_{\text{diff}}\right\}\right)$. For notational convenience, denote

$$r_{\max} := \min\left\{\sigma \min\{1, \beta/2\rho\}, r_{\text{diff}}\right\},$$

which is a constant. As $f + {}^{\mu}\!\!\downarrow$ is a global underestimator of and approximates the function $f + \frac{\beta}{2}\|\cdot\|^2 + \iota$ with arbitrary precision as $\mu \to 0$, the previous statement and [12, Theorem 1.25] imply that there exist some $0 < \mu_{\max} \leq 1/\beta$ such that for any $\mu \in (0, \mu_{\max}]$, the function $f + {}^{\mu}\!\!\downarrow$ will achieve a local minimum x_μ over $B(\bar{x}; r_{\max})$ where $\nabla(f + {}^{\mu}\!\!\downarrow)$ vanishes, *i.e.*,

$$\nabla(f + {}^{\mu}\!\!\downarrow)(x_\mu) = \nabla f(x_\mu) + \beta x_\mu + \frac{1}{\mu}\left(x_\mu - \mathbf{\Pi}\left(x_\mu\right)\right) = 0 \tag{24}$$

$$\Rightarrow x_\mu = \frac{1}{\beta\mu + 1}\left(\mathbf{\Pi}(x_\mu) - \mu\nabla f(x_\mu)\right). \tag{25}$$

As the right hand side of the last equation is a singleton, this minimum must be unique. Finally to show the smoothness $f + {}^{\mu}\!\!\downarrow$, for any $x \in B(\bar{x}; r_{\max})$, we have

$$\begin{aligned}
\nabla\left(f + {}^{\mu}\!\!\downarrow\right)(x) &= \nabla\left(f + \frac{\beta}{2}\|\cdot\|^2 + {}^{\mu}\iota\right)(x) \\
&= \nabla f(x) + \beta x + \frac{1}{2\mu}\nabla d^2(x) \\
&\overset{a)}{=} \nabla f(x) + \beta x + \frac{1}{\mu}\left(x - \mathbf{\Pi}(x)\right) \\
&= \nabla f(x) + \left(\beta + \frac{1}{\mu}\right)x - \frac{1}{\mu}\mathbf{\Pi}(x),
\end{aligned} \tag{26}$$

where a) uses Lemma 6. Thus, for any $x_1, x_2 \in B(\bar{x}; r_{\max})$ we have

$$\begin{aligned}
&\left\|\nabla\left(f + \frac{\beta}{2}\|\cdot\|^2 + {}^{\mu}\iota\right)(x_1) - \nabla\left(f + \frac{\beta}{2}\|\cdot\|^2 + {}^{\mu}\iota\right)(x_2)\right\| \\
&= \left\|\nabla f(x_1) + \left(\beta + \frac{1}{\mu}\right)x_1 - \frac{1}{\mu}\mathbf{\Pi}(x_1) - \nabla f(x_2) - \left(\beta + \frac{1}{\mu}\right)x_2 + \frac{1}{\mu}\mathbf{\Pi}(x_2)\right\|
\end{aligned}$$

$$\overset{a)}{\leq} \underbrace{\|\nabla f(x_1) - \nabla f(x_2)\|}_{\leq L\|x_1 - x_2\|} + \left(\beta + \frac{1}{\mu}\right)\|x_1 - x_2\| + \frac{1}{\mu}\underbrace{\|\mathbf{\Pi}(x_1) - \mathbf{\Pi}(x_2)\|}_{\leq \widetilde{L}\|x_1 - x_2\|}$$
$$\leq \left(L + \beta + \frac{1}{\mu} + \widetilde{L}\right)\|x_1 - x_2\|,$$

where we have used the following in a): ∇f is L-Lipschitz everywhere due to f being an $L-$smooth function in $\mathbf{E}$ ([11, Theorem 18.15]), and $\mathbf{\Pi}$ is $\widetilde{L}$-Lipschitz continuous on $B(\bar{x}; r_{\max})$, as shown in step 1. This completes the proof for (i).

(ii): Using [12, Theorem 1.25], as $\mu \to 0$, we have $x_\mu \to \bar{x}$, and $(f + {}^{\mu}\mathbb{1})(x_\mu) \to f(\bar{x}) + \frac{\beta}{2}\|\bar{x}\|^2$. Note that x_μ reaches $\bar{x}$ only in limit, as otherwise Assumption 2 will be violated.

B.3 Proof to Proposition 2

B.3.1 Proof to Proposition 2(i)

We will use the following definition.

Definition 5 (Resolvent and reflected resolvent [11, pages 333, 336])**.** For a lower-semicontinuous, proper, and convex function h, the resolvent and reflected resolvent of its subdifferential operator are defined by $\mathbb{J}_{\gamma\partial h} = (\mathbb{I} + \gamma\partial h)^{-1}$ and $\mathbb{R}_{\gamma\partial h} = 2\mathbb{J}_{\gamma\partial h} - \mathbb{I}$, respectively.

The proof of (i) is proven in two steps. First, we show that the reflection operator of $\mathbb{T}_\mu$, defined by

$$\mathbb{R}_\mu = 2\mathbb{T}_\mu - \mathbb{I}, \tag{27}$$

is contractive on $B(\bar{x}, r_{\max})$, and using this we show that $\mathbb{T}_\mu$ in also contractive there in the second step. To that goal, note that $\mathbb{R}_\mu$ can be represented as:

$$\mathbb{R}_\mu = \left(2\mathbf{prox}_{\gamma\,{}^{\mu}\mathbb{1}} - \mathbb{I}\right)\left(2\mathbf{prox}_{\gamma f} - \mathbb{I}\right), \tag{28}$$

which can be proven by simply using (7) and (27) on the left-hand side and by expanding the factors on the right-hand side. Now, the operator $2\mathbf{prox}_{\gamma f} - \mathbb{I}$ associated with the α-strongly convex and L-smooth function f is a contraction mapping for any $\gamma > 0$ with the contraction factor

$$\kappa = \max\left\{\frac{\gamma L - 1}{\gamma L + 1}, \frac{1 - \gamma\alpha}{\gamma\alpha + 1}\right\} \in (0, 1),$$

which follows from [17, Theorem 1].

Next, we show that $2\mathbf{prox}_{\gamma\,{}^{\mu}\mathbb{1}} - \mathbb{I}$ is nonexpansive on $B(\bar{x}; r_{\max})$ for any $\mu \in (0, \mu_{\max}]$. For any $\mu \in (0, \mu_{\max}]$, define the function g as follows:

$$g(y) = \begin{cases} {}^{\mu}\mathbb{1}(y), & \text{if } y \in B(\bar{x}; r_{\max}) \\ \liminf_{\tilde{y}\to y} {}^{\mu}\mathbb{1}(\tilde{y}), & \text{if } \|y - \bar{x}\| = r_{\max} \\ +\infty, & \text{else.} \end{cases}$$

which is lower-semicontinuous, proper, and convex everywhere due to [11, Lemma 1.31 and Corollary 9.10]. As a result for $\mu \in (0, \mu_{\max}]$, we have $\mathbf{prox}_{\gamma g} = \mathbb{J}_{\gamma\partial g}$ on $\mathbf{E}$ and $\mathbf{prox}_{\gamma g}$ is firmly

nonexpansive and single-valued everywhere, which follows from [11, Proposition 12.27, Proposition 16.34, and Example 23.3]. But, for $y \in B(\bar{x}; r_{\max})$, we have ${}^{\mu}\mathbb{0}(y) = g(y)$ and $\nabla {}^{\mu}\mathbb{0}(y) = \partial g(y)$. Thus, on $B(\bar{x}; r_{\max})$, the operator $\mathbf{prox}_{\gamma\, {}^{\mu}\mathbb{0}} = \mathbb{J}_{\gamma \nabla\, {}^{\mu}\mathbb{0}}$, and it is firmly nonexpansive and single-valued for $\mu \in (0, \mu_{\max}]$. Any firmly nonexpansive operator $\mathbb{A}$ has a nonexpansive reflection operator $2\mathbb{A} - \mathbb{I}$ on its domain of firm nonexpansiveness [11, Proposition 4.2]. Hence, on $B(\bar{x}; r_{\max})$, for $\mu \in (0, \mu_{\max}]$ the operator $2\mathbf{prox}_{\gamma\, {}^{\mu}\mathbb{0}} - \mathbb{I}$ is nonexpansive using (28).

Now we show that $\mathbb{R}_\mu$ is contractive for every $x_1, x_2 \in B(\bar{x}; r_{\max})$ and $\mu \in (0, \mu_{\max}]$, we have

$$\begin{aligned}
&\|\mathbb{R}_\mu(x_1) - \mathbb{R}_\mu(x_2)\| \\
&= \left\| \left(\left(2\mathbf{prox}_{\gamma\, {}^{\mu}\mathbb{0}} - \mathbb{I} \right) \left(2\mathbf{prox}_{\gamma f} - \mathbb{I} \right) \right)(x_1) - \left(\left(2\mathbf{prox}_{\gamma\, {}^{\mu}\mathbb{0}} - \mathbb{I} \right) \left(2\mathbf{prox}_{\gamma f} - \mathbb{I} \right) \right)(x_2) \right\| \qquad (29) \\
&\leq \left\| \left(2\mathbf{prox}_{\gamma f} - \mathbb{I} \right)(x_1) - \left(2\mathbf{prox}_{\gamma f} - \mathbb{I} \right)(x_2) \right\| \leq \kappa \|x_1 - x_2\|,
\end{aligned}$$

where the last inequality uses κ-contractiveness of $2\mathbf{prox}_{\gamma f} - \mathbb{I}$ thus proving that $\mathbb{R}_\mu$ acts as a contractive operator on $B(\bar{x}; r_{\max})$ for $\mu \in (0, \mu_{\max}]$. Similarly, for any $x_1, x_2 \in B(\bar{x}; r_{\max})$, using $(\mathcal{A}_\mu)$ and the triangle inequality we have

$$\|\mathbb{T}_\mu(x_1) - \mathbb{T}_\mu(x_2)\| \leq \left(\frac{1+\kappa}{2} \right) \|x_1 - x_2\|, \qquad (30)$$

and as $\kappa' = (1+\kappa)/2 \in [0, 1)$; the operator $\mathbb{T}_\mu$ is $\kappa'-$contractive on on $B(\bar{x}; r_{\max})$, for $\mu \in (0, \mu_{\max}]$.

B.3.2 Proof to Proposition 2(ii)

Recalling $\mathbb{T}_\mu = (1/2)\mathbb{R}_\mu + (1/2)\mathbb{I}$ from (27), using (28), and then expanding, and finally using Lemma 5 and triangle inequality, we have for any $\mu, \tilde{\mu} \in (0, \mu_{\max}]$, $x \in B(\bar{x}; r_{\max})$,

$$\begin{aligned}
\|\mathbb{T}_\mu(x) - \mathbb{T}_{\tilde{\mu}}(x)\| &= \|\mathbf{prox}_{\gamma\, {}^{\mu}\mathbb{0}} y - \mathbf{prox}_{\gamma\, {}^{\tilde{\mu}}\mathbb{0}} y\| \\
&\overset{a)}{=} \left\| \frac{\mu}{\gamma + \mu(\beta\gamma+1)} y + \frac{\gamma}{\gamma + \mu(\beta\gamma+1)} \mathbf{\Pi}\left(\frac{y}{\beta\gamma+1} \right) \right. \\
&\quad \left. - \frac{\tilde{\mu}}{\gamma + \tilde{\mu}(\beta\gamma+1)} y - \frac{\gamma}{\gamma + \tilde{\mu}(\beta\gamma+1)} \mathbf{\Pi}\left(\frac{y}{\beta\gamma+1} \right) \right\| \\
&\leq \left\| \left(\frac{\mu}{\gamma + \mu(\beta\gamma+1)} - \frac{\tilde{\mu}}{\gamma + \tilde{\mu}(\beta\gamma+1)} \right) \right\| \|y\| \\
&\quad + \left\| \left(\frac{\gamma}{\gamma + \mu(\beta\gamma+1)} - \frac{\gamma}{\gamma + \tilde{\mu}(\beta\gamma+1)} \right) \right\| \left\| \mathbf{\Pi}\left(\frac{y}{\beta\gamma+1} \right) \right\|. \qquad (31)
\end{aligned}$$

Now, in (31), the coefficient of $\|y\|$ satisfies

$$\begin{aligned}
\left\| \frac{\mu}{\gamma + \mu(\beta\gamma+1)} - \frac{\tilde{\mu}}{\gamma + \tilde{\mu}(\beta\gamma+1)} \right\| &= \left\| \frac{\frac{1}{\gamma}(\mu - \tilde{\mu})}{\left(1 + \mu(\beta + \frac{1}{\gamma})\right)\left(1 + \tilde{\mu}(\beta + \frac{1}{\gamma})\right)} \right\| \\
&\leq \frac{1}{\gamma} \|\mu - \tilde{\mu}\|,
\end{aligned}$$

and similarly the coefficient of $\| \mathbf{\Pi}\left(y/(\beta\gamma+1)\right) \|$ satisfies

$$\left\| \left(\frac{\gamma}{\gamma + \mu(\beta\gamma+1)} - \frac{\gamma}{\gamma + \tilde{\mu}(\beta\gamma+1)} \right) \right\| = \left\| - \frac{\gamma(\beta\gamma+1)(\mu - \tilde{\mu})}{(\gamma + \mu(\beta\gamma+1))\,(\gamma + \tilde{\mu}(\beta\gamma+1))} \right\|$$

$$\leq \left(\beta + \frac{1}{\gamma}\right) \|\mu - \tilde{\mu}\|.$$

Putting the last two inequalities in (31), and then replacing $y = (2\mathbf{prox}_{\gamma f} - \mathbb{I})(x)$, we have for any $x \in \mathcal{B}$, and for any $\mu, \tilde{\mu} \in \mathbf{R}_{++}$,

$$\begin{aligned}
&\|\mathbb{T}_\mu(x) - \mathbb{T}_{\tilde{\mu}}(x)\| \\
\leq &\frac{1}{\gamma} \|\mu - \tilde{\mu}\| \, \|y\| + \left(\beta + \frac{1}{\gamma}\right) \|\mu - \tilde{\mu}\| \left\| \mathbf{\Pi}\left(\frac{y}{\beta\gamma + 1}\right)\right\| \qquad (32)\\
\overset{a)}{=} &\left(\frac{1}{\gamma}\|2\mathbf{prox}_{\gamma f}(x) - x\| + \left(\beta + \frac{1}{\gamma}\right) \left\| \mathbf{\Pi}\left(\frac{2\mathbf{prox}_{\gamma f}(x) - x}{\beta\gamma + 1}\right)\right\|\right) \|\mu - \tilde{\mu}\|. \qquad (33)
\end{aligned}$$

Now, as $B(\bar{x}; r_{\max})$ is a bounded set and $x \in \mathcal{B}$, norm of the vector $y = 2\mathbf{prox}_{\gamma f}(x) - x$ can be upper-bounded over $B(\bar{x}; r_{\max})$ because $2\mathbf{prox}_{\gamma f} - \mathbb{I}$ is continuous (in fact contractive) as shown in (i). Similarly, $\| \mathbf{\Pi}\left((2\mathbf{prox}_{\gamma f}(x) - x)/(\beta\gamma + 1)\right) \|$ can be upper-bounded $B(\bar{x}; r_{\max})$. Combining the last two-statements, it follows that there exists some $\ell > 0$ such that

$$\sup_{x \in B(\bar{x}; r_{\max})} \left(\frac{1}{\gamma}\|2\mathbf{prox}_{\gamma f}(x) - x\| + \left(\beta + \frac{1}{\gamma}\right) \left\| \mathbf{\Pi}\left(\frac{2\mathbf{prox}_{\gamma f}(x) - x}{\beta\gamma + 1}\right)\right\|\right) \leq \ell,$$

and putting the last inequality in (33), we arrive at the claim.

B.4 Proof to Proposition 3

The structure of the proof follows that of [11, Proposition 25.1(ii)]. Let $\mu \in (0, \mu_{\max}]$. Recalling Definition 5, and due to Proposition 1(i), $x_\mu \in B(\bar{x}; r_{\max})$ satisfies

$$\begin{aligned}
x_\mu = &\operatorname*{argmin}_{B(\bar{x}; r_{\max})} f(x) + {}^{\mu}\mathbb{1}(x) = \mathbf{zer}(\nabla f + \nabla\, {}^{\mu}\mathbb{1})\\
\Leftrightarrow &(\exists y \in \mathbf{E})\ x_\mu - y \in \gamma \nabla\, {}^{\mu}\mathbb{1}(x_\mu) \text{ and } y - x_\mu \in \gamma \nabla f(x_\mu)\\
\Leftrightarrow &\,(\exists y \in \mathbf{E})\ 2x_\mu - y \in (\mathbb{I} + \gamma \nabla\, {}^{\mu}\mathbb{1})\,(x_\mu) \text{ and } y \in (\mathbb{I} + \gamma \nabla f)\,(x_\mu)\\
\Leftrightarrow &\,(\exists y \in \mathbf{E})\ \underbrace{(\mathbb{I} + \gamma \nabla\, {}^{\mu}\mathbb{1})^{-1}}_{=\mathbb{J}_{\gamma \nabla\, {}^{\mu}\mathbb{1}}}(2x_\mu - y) \ni x_\mu \text{ and } \underbrace{(\mathbb{I} + \gamma \nabla f)^{-1}}_{=\mathbb{J}_{\gamma \nabla f}}(y) \ni x_\mu\\
\overset{a)}{\Leftrightarrow} &\,(\exists y \in \mathbf{E})\ x_\mu \in \mathbb{J}_{\gamma \nabla\, {}^{\mu}\mathbb{1}}(2x_\mu - y) \text{ and } x_\mu = \mathbb{J}_{\gamma \nabla f}(y)\\
\overset{b)}{\Leftrightarrow} &\,(\exists y \in \mathbf{E})\ x_\mu = \mathbb{J}_{\gamma \nabla\, {}^{\mu}\mathbb{1}}\mathbb{R}_{\gamma \nabla f}(y) \text{ and } x_\mu = \mathbb{J}_{\gamma \nabla f}(y), \qquad (34)
\end{aligned}$$

where $a)$ uses the facts (shown in the proof to Proposition 2) that : (i) $\mathbb{J}_{\gamma \nabla f}$ is a single-valued operator everywhere, whereas $\mathbb{J}_{\gamma \nabla\, {}^{\mu}\mathbb{1}}$ is a single-valued operator on the region of convexity $B(\bar{x}; r_{\max})$, and $b)$ uses the observation that $x_\mu = \mathbb{J}_{\gamma \nabla f}(y)$ can be expressed as

$$x_\mu = \mathbb{J}_{\gamma \nabla f}(y) \Leftrightarrow 2x_\mu - y = (2\mathbb{J}_{\gamma \nabla f} - \mathbb{I})\, y = \mathbb{R}_{\gamma \nabla f}(y). \qquad (35)$$

Also, using the last expression, we can write the first term of (34) as

$$\begin{aligned}
&\mathbb{J}_{\gamma \nabla\, {}^{\mu}\mathbb{1}}\mathbb{R}_{\gamma \nabla f}(y) = x_\mu\\
\Leftrightarrow &2\mathbb{J}_{\gamma \nabla\, {}^{\mu}\mathbb{1}}\mathbb{R}_{\gamma \nabla f}(y) - y \overset{a)}{=} \mathbb{R}_{\gamma \nabla f}(y)
\end{aligned}$$

$$
\begin{aligned}
\Leftrightarrow y &= 2\mathbb{J}_{\gamma\nabla\, \mu_{\mathbb{0}}}\mathbb{R}_{\gamma\nabla f}(y) - \mathbb{R}_{\gamma\nabla f}(y) \\
&= \left(2\mathbb{J}_{\gamma\nabla\, \mu_{\mathbb{0}}} - \mathbb{I}\right)\left(\mathbb{R}_{\gamma\nabla f}(y)\right) \\
&= \mathbb{R}_{\gamma\nabla\, \mu_{\mathbb{0}}}\mathbb{R}_{\gamma\nabla f}(y) \\
\Leftrightarrow y &\in \mathbf{fix}\left(\mathbb{R}_{\gamma\nabla\, \mu_{\mathbb{0}}}\mathbb{R}_{\gamma\nabla f}\right), \qquad (36)
\end{aligned}
$$

where a) uses (35). Because for lower-semicontinuous, proper, and convex function, the resolvent of the subdifferential is equal to its proximal operator [11, Proposition 12.27, Proposition 16.34, and Example 23.3], we have $\mathbb{J}_{\gamma\partial f} = \mathbf{prox}_{\gamma f}$ with both being single-valued. Using the last fact along with (34), (36) we have

$$
\begin{aligned}
& x_\mu \in \mathbf{zer}\left(\nabla f + \nabla\, \mu_{\mathbb{0}}\right) \\
\Leftrightarrow & \left(\exists y \in \mathbf{E}\right)\ y \in \mathbf{fix}\left(\mathbb{R}_{\gamma\nabla\, \mu_{\mathbb{0}}}\mathbb{R}_{\gamma\nabla f}\right) \text{ and } x_\mu = \mathbf{prox}_{\gamma f}(y) \\
\Leftrightarrow & x_\mu \in \mathbf{prox}_{\gamma f}\left(\mathbf{fix}\left(\mathbb{R}_{\gamma\nabla\, \mu_{\mathbb{0}}}\mathbb{R}_{\gamma\partial f}\right)\right),
\end{aligned}
$$

but x_μ is unique due to Proposition 1, so the inclusion can be replaced with equality. Thus x_μ, satisfies

$$
x_\mu = \mathbf{zer}\left(\nabla f + \nabla\, \mu_{\mathbb{0}}\right) = \mathbf{prox}_{\gamma f}\left(\mathbf{fix}\left(\mathbb{R}_{\gamma\nabla\, \mu_{\mathbb{0}}}\mathbb{R}_{\gamma\partial f}\right)\right),
$$

where the sets are singletons due to Proposition 1 and single-valuedness of $\mathbf{prox}_{\gamma f}$. Also, because $\mathbb{T}_\mu$ in (7) and $\mathbb{R}_\mu$ in (27) have the same fixed point set (follows from (27)), using (28) we get

$$
\mathbf{fix}\,\mathbb{T}_\mu = \mathbf{fix}\,\mathbb{R}_\mu = \mathbf{fix}\left(\mathbb{R}_{\gamma\nabla\, \mu_{\mathbb{0}}}\mathbb{R}_{\gamma\partial f}\right),
$$

hence we have arrived at (8).

B.5 Proof to Lemma 1

(i): This follows directly from the proof to Proposition 1.

(ii): From (9), and recalling that $\eta' > 1$, for any $\mu \in (0, \mu_{\max}]$, we have (10).

Recalling Definition 5, and using the fact that for lower-semicontinuous, proper, and convex function, the resolvent of the subdifferential is equal to its proximal operator [11, Proposition 12.27, Proposition 16.34, and Example 23.3], we have $\mathbb{J}_{\gamma\partial f} = \mathbf{prox}_{\gamma f}$ with both being single-valued. So, from Proposition 3, the fixed point z_μ corresponding to x_μ satisfies

$$
x_\mu = \mathbf{prox}_{\gamma f}(z_\mu) = \left(\mathbb{I} + \gamma\partial f\right)^{-1}(z_\mu) \Leftrightarrow z_\mu = x_\mu + \gamma\nabla f(x_\mu). \qquad (37)
$$

Hence, for any $\mu \in (0, \mu_{\max}]$

$$
\begin{aligned}
\|z_\mu - \bar{x}\| &= \|x_\mu + \gamma\nabla f(x_\mu) - \bar{x}\| \\
&\leq \|x_\mu - \bar{x}\| + \gamma\|\nabla f(x_\mu)\| \\
\Leftrightarrow r_{\max} - \|z_\mu - \bar{x}\| &\geq r_{\max} - \|x_\mu - \bar{x}\| - \gamma\|\nabla f(x_\mu)\| \\
&\overset{a)}{\geq} \frac{\eta' - 1}{\eta'} r_{\max} - \gamma\|\nabla f(x_\mu)\|,
\end{aligned}
$$

where a) uses (10). Because, for the strongly convex and smooth function f, its gradient is bounded over a bounded set $B(\bar{x}; r_{\max})$ [13, Lemma 1, §1.4.2], then for γ satisfying (13) and the definition of ψ in (12), we have (11) for any $\mu \in (0, \mu_{\max}]$. To prove (14), note that

$$\begin{aligned}
&\lim_{\mu\to 0} (r_{\max} - \|z_\mu - \bar{x}\|) - \psi \\
\overset{a)}{=} &\lim_{\mu\to 0} (r_{\max} - \|x_\mu + \gamma\nabla f(x_\mu) - \bar{x}\|) - \frac{\eta' - 1}{\eta'} r_{\max} + \gamma \max\nolimits_{x\in B(\bar{x}; r_{\max})} \|\nabla f(x)\| \\
\overset{b)}{=} &(r_{\max} - \|\bar{x} + \gamma\nabla f(\bar{x}) - \bar{x}\|) - \frac{\eta' - 1}{\eta'} r_{\max} + \gamma \max\nolimits_{x\in B(\bar{x}; r_{\max})} \|\nabla f(x)\| \\
= &\frac{1}{\eta'} r_{\max} + \gamma \left(\max\nolimits_{x\in B(\bar{x}; r_{\max})} \|\nabla f(x)\| - \|\nabla f(\bar{x})\|\right) > 0,
\end{aligned} \tag{38}$$

where in a) we have used (37) and (12), in b) we have used smoothness of f along with Proposition 1(ii). Inequality (38) along with (11) implies (14).

B.6 Proof to Theorem 1

We use the following result from [41] in proving Theorem 1.

Theorem 2 (Convergence of local contraction mapping [41, pp. 313-314])**.** *Let $\mathbb{A} : \mathbf{E} \to \mathbf{E}$ be some operator. If there exist $\tilde{x}$, $\omega \in (0, 1)$, and $r > 0$ such that*

(a) the operator $\mathbb{A}$ is ω-contractive on $B(\tilde{x}; r)$, i.e., for all x_1, x_2 in $B(\tilde{x}; r)$:

$$\|\mathbb{A}(x_1) - \mathbb{A}(x_2)\| \leq \omega \|x_1 - x_2\|,$$

and

(b)

$$\|\mathbb{A}(\tilde{x}) - \tilde{x}\| \leq (1 - \omega) r.$$

Then $\mathbb{A}$ has a unique fixed point in $B(\tilde{x}; r)$ and the iteration scheme $x_{n+1} = \mathbb{A}(x_n)$ with the initialization $x_0 := \tilde{x}$ linearly converges to that unique fixed point.

Furthermore, recall that NExOS (Algorithm 1) can be compactly represented using ($\mathcal{A}_\mu$) as follows. For any $m \in \{1, 2, \ldots, N\}$ (equivalently for each $\mu_m \in \{\mu_1, \ldots, \mu_N\}$),

$$z_{\mu_m}^{n+1} = \mathbb{T}_{\mu_m}\left(z_{\mu_m}^{n}\right), \tag{39}$$

where $z_{\mu_m}^0$ is initialized at $z_{\mu_{m-1}}$.

Now we start the proof. From Proposition 2 and Remark 1, for any $\mu \in \mathfrak{M}$, the operator $\mathbb{T}_\mu$ is a κ'-contraction mapping over the region of convexity $B(\bar{x}; r_{\max})$, where $\kappa' \in (0, 1)$. From Proposition 1, there will be a unique local minimum x_μ of over $B(\bar{x}; r_{\max})$. Suppose, instead of the exact fixed point $z_{\mu_{m-1}} \in \mathbf{fix}\, \mathbb{T}_{\mu_{m-1}}$, we have computed $\widetilde{z}$, which is an ϵ-approximate fixed point of $\mathbb{T}_{\mu_{m-1}}$ in $B(\bar{x}; r_{\max})$, i.e.,

$$\|\widetilde{z} - \mathbb{T}_{\mu_{m-1}}(\widetilde{z})\| \leq \epsilon, \tag{40}$$

where ϵ satisfies: Then, we have:

$$\|\mathbb{T}_{\mu_{m-1}}(\widetilde{z}) - z_{\mu_{m-1}}\| = \|\mathbb{T}_{\mu_{m-1}}(\widetilde{z}) - \mathbb{T}_{\mu_{m-1}}(z_{\mu_{m-1}})\| \overset{a)}{\leq} \kappa' \underbrace{\|\widetilde{z} - z_{\mu_{m-1}}\|}_{\leq \epsilon} \leq \epsilon, \tag{41}$$

where $a)$ uses κ'-contractive nature of $\mathbb{T}_{\mu_{m-1}}$ over $B(\bar{x}; r_{\max})$. Hence, using triangle inequality,

$$\begin{aligned}\|\widetilde{z} - \bar{x}\| &\overset{a)}{\leq} \|\widetilde{z} - \mathbb{T}_{\mu_{m-1}}(\widetilde{z})\| + \|\mathbb{T}_{\mu_{m-1}}(\widetilde{z}) - z_{\mu_{m-1}}\| + \|z_{\mu_{m-1}} - \bar{x}\| \\ &\overset{b)}{\leq} 2\epsilon + \|z_{\mu_{m-1}} - \bar{x}\|,\end{aligned}$$

where $a)$ uses triangle inequality and $b)$ uses (41). As $\epsilon \in \left[0, \overline{\epsilon}\right)$, where $\overline{\epsilon}$ is defined in (15), due to (11), we have

$$r_{\max} - \|\widetilde{z} - \bar{x}\| > \psi. \tag{42}$$

Define $\Delta = \left((1-\kappa')\psi - \epsilon\right)/\ell$, which will be positive due to $\epsilon \in \left[0, \overline{\epsilon}\right)$ and (15). Next, select $\theta \in (0,1)$ such that $\overline{\Delta} = \theta\Delta < \mu_1$, hence there exists a $\rho \in (0,1)$ such that $\overline{\Delta} = (1-\rho)\mu_1$. Now reduce the penalty parameter using the following rule:

$$\begin{aligned}\mu_2 &= \mu_1 - \overline{\Delta} = \rho\mu_1, \\ \mu_3 &= \mu_2 - \rho\overline{\Delta} = \mu_2 - \rho(1-\rho)\mu_1 = \mu_2 - (1-\rho)\mu_2 = \rho\mu_2 = \rho^2\mu_1 \\ \mu_4 &= \mu_3 - \rho^2\overline{\Delta} = \mu_3 - \rho^2(1-\rho)\mu_1 = \mu_3 - (1-\rho)\mu_3 = \rho\mu_3 = \rho^3\mu_1 \\ &\vdots \\ \mu_m &= \mu_{m-1} - \rho^{m-2}\overline{\Delta} = \rho\mu_{m-1} = \rho^{m-1}\mu_1.\end{aligned} \tag{43}$$

Next, we initialize the iteration scheme $z^{n+1}_{\mu_m} = \mathbb{T}_{\mu_m}\left(z^n_{\mu_m}\right)$ at $z^0_{\mu_m} := \widetilde{z}$. Around this initial point, let us consider the open ball $B(\widetilde{z}, \psi)$. For any $x \in B(\widetilde{z}; \psi)$, we have

$$\|x - \bar{x}\| \leq \underbrace{\|x - \widetilde{z}\|}_{<\psi} + \|\widetilde{z} - \bar{x}\| < \psi + \|\widetilde{z} - \bar{x}\| < r_{\max},$$

where the last inequality follows from (42). Thus we have shown that $B(\widetilde{z}; \psi) \subseteq B(\bar{x}; r_{\max})$. Hence, from Proposition 2, on $B(\widetilde{z}; \psi)$, the Douglas-Rachford operator $\mathbb{T}_{\mu_m}$ is contractive. Next, we have $\|\mathbb{T}_{\mu_m}(\widetilde{z}) - \widetilde{z}\| \leq (1-\kappa')\psi$, because

$$\begin{aligned}\|\mathbb{T}_{\mu_m}(\widetilde{z}) - \widetilde{z}\| &= \|\mathbb{T}_{\mu_m}(\widetilde{z}) - \mathbb{T}_{\mu_{m-1}}(\widetilde{z}) + \mathbb{T}_{\mu_{m-1}}(\widetilde{z}) - \widetilde{z}\| \\ &\overset{a)}{\leq} \|\mathbb{T}_{\mu_m}(\widetilde{z}) - \mathbb{T}_{\mu_{m-1}}(\widetilde{z})\| + \|\mathbb{T}_{\mu_{m-1}}(\widetilde{z}) - \widetilde{z}\| \\ &\overset{b)}{\leq} \ell\|\mu_m - \mu_{m-1}\| + \epsilon \\ &\overset{c)}{\leq} \epsilon + \ell\Delta \\ &\overset{d)}{\leq} (1-\kappa')\psi,\end{aligned}$$

where a) triangle inequality, b) uses Proposition 2(ii) and (40), c) uses (43), and d) uses $\|\mu_m - \mu_{m-1}\| \leq \overline{\Delta} \leq \Delta$. Thus, both conditions of Theorem 2 are satisfied, and $z^n_{\mu_m}$ in (39) will linearly converge to the unique fixed point $z_{\mu_m}$of the operator $\mathbb{T}_{\mu_m}$, and $x^n_{\mu_m}, y^n_{\mu_m}$ will linearly converge to x_{μ_m}. This completes the proof.

B.7 Proof to Lemma 2

First, we show that, for the given initialization of z_{init}, the iterates $z^n_{\mu_1}$ stay in $\overline{B}(z_{\mu_1}; \|z_{\text{init}} - z_{\mu_1}\|)$ for any $n \in \mathbf{N}$ via induction. The base case is true via given. Let, $z^n_{\mu_1} \in \overline{B}(z_{\mu_1}; \|z_{\text{init}} - z_{\mu_1}\|)$, *i.e.*,

$$\|z^n_{\mu_1} - z_{\mu_1}\| \leq \|z_{\text{init}} - z_{\mu_1}\|. \tag{44}$$

Then,

$$\begin{aligned} \left\|z^{n+1}_{\mu_1} - z_{\mu_1}\right\| &\overset{a)}{=} \left\|\mathbb{T}_{\mu_1}\left(z^n_{\mu_1}\right) - \mathbb{T}_{\mu_1}\left(z_{\mu_1}\right)\right\| \\ &\overset{b)}{\leq} \kappa' \|z^n_{\mu_1} - z_{\mu_1}\|, \\ &\overset{c)}{\leq} \kappa' \|z_{\text{init}} - z_{\mu_1}\| \\ &\leq \|z_{\text{init}} - z_{\mu_1}\|, \end{aligned} \tag{45}$$

where a) uses $z_{\mu_1} \in \mathbf{fix}\, \mathbb{T}_\mu$, and b) uses Proposition 2, and c) uses (44). So, the iterates $z^n_{\mu_1}$ stay in $\overline{B}(z_{\mu_1}; \|z_{\text{init}} - z_{\mu_1}\|)$. As, $\kappa' \in (0, 1)$, (45) also implies that z^n_μ linearly converges to z_μ with the rate of at least κ'. Then using similar reasoning presented in the proof to Theorem 1, we have x^n_μ and y^n_μ linearly converge to the unique local minimum x_μ of ($\mathcal{P}_\mu$). This completes the proof.