Signal-Recovery Methods for Compressive Sensing Using Nonconvex Sparsity-Promoting Functions

by

Flávio C. A. Teixeira
B.Sc., Centro Universitário de Brasília, 2005
M.Sc., Universidade de Brasília, 2008

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Signal-Recovery Methods for Compressive Sensing Using Nonconvex Sparsity-Promoting Functions

by

Flávio C. A. Teixeira
B.Sc., Centro Universitário de Brasília, 2005
M.Sc., Universidade de Brasília, 2008

Supervisory Committee

Dr. Andreas Antoniou, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Stuart W. A. Bergen, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Jane Ye, Outside Member
(Department of Mathematics and Statistics, University of Victoria)

Supervisory Committee

Dr. Andreas Antoniou, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Stuart W. A. Bergen, Co-Supervisor
(Department of Electrical and Computer Engineering, University of Victoria)

Dr. Jane Ye, Outside Member
(Department of Mathematics and Statistics, University of Victoria)

ABSTRACT

Recent research has shown that compressible signals can be recovered from a very limited number of measurements by minimizing nonconvex functions that closely resemble the ℓ_0-norm function. These functions have sparse minimizers and, therefore, are called sparsity-promoting functions (SPFs). Recovery is achieved by solving a nonconvex optimization problem when using these SPFs. Contemporary methods for the solution of such difficult problems are inefficient and not supported by robust convergence theorems.

New signal-recovery methods for compressive sensing that can be used to solve nonconvex problems efficiently are proposed. Two categories of methods are considered, namely, sequential convex formulation (SCF) and proximal-point (PP) based methods. In SCF methods, quadratic or piecewise-linear approximations of the SPF are employed. Recovery is achieved by solving a sequence of convex optimization problems efficiently with state-of-the-art solvers. Convex problems are formulated as regularized least-squares, second-order cone programming, and weighted ℓ_1-norm minimization problems. In PP based methods, SPFs that entail rich optimization properties are employed. Recovery is achieved by iteratively performing two fundamental operations, namely, computation of the PP of the SPF and projection of the PP onto a convex set. The first operation is performed analytically or numerically

by using a fast iterative method. The second operation is performed efficiently by computing a sequence of closed-form projectors.

The proposed methods have been compared with the leading state-of-the-art signal-recovery methods, namely, the gradient-projection method of Figueiredo, Nowak, and Wright, the ℓ_1-LS method of Kim, Koh, Lustig, Boyd, and Gorinevsky, the ℓ_1-Magic method of Candès and Romberg, the spectral projected-gradient ℓ_1-norm method of Berg and Friedlander, the iteratively reweighted least squares method of Chartrand and Yin, the difference-of-two-convex-functions method of Gasso, Rakotomamonjy, and Canu, and the NESTA method of Becker, Bobin, and Candès. The comparisons concerned the capability of the proposed and competing algorithms in recovering signals in a wide range of test problems and also the computational efficiency of the various algorithms.

Simulation results demonstrate that improved reconstruction performance, measurement consistency, and comparable computational cost are achieved with the proposed methods relative to the competing methods. The proposed methods are robust, are supported by known convergence theorems, and lead to fast convergence. They are, as a consequence, particularly suitable for the solution of hard recovery problems of large size that entail large dynamic range and, are, in effect, strong candidates for use in many real-world applications.

Contents

List of Abbreviations

AP	alternating projection
BCQP	bound-constrained quadratic programming
BP	basis pursuit
CC	computational cost
CS	compressive sensing
DC	difference-of-two-convex-functions
DCT	discrete cosine transform
DFT	discrete Fourier transform
DR	dynamic range
FFT	fast Fourier transform
FIPPP	*fast* iterative proximal-point projection
FISTA	fast iterative shrinkage-thresholding algorithm
FOS	first-order solver
GPSR	gradient projection for sparse reconstruction
i.i.d.	independent and identically distributed
IPPP	iterative proximal-point projection
IRWL1	iterative reweighted ℓ_1
IRWLS	iteratively reweighted least squares
KKT	Karush-Kuhn-Tucker
LASSO	least absolute shrinkage and selection operator
LP	linear programming
MC	measurement consistency
MDP	monotonic decreasing property
ME	Moreau envelope
MM	majorization-minimization
MRF	minimum required fraction

PCG	preconditioned conjugate gradient
PLA	piecewise-linear approximation
PP	proximal-point
PPR	probability of perfect recovery
QA	quadratic approximation
QP	quadratic programming
RLS	regularized least-squares
RP	reconstruction performance
SCAD	smoothly-clipped absolute deviation
SCF	sequential convex formulation
SeDuMi	self-dual-minimization
SOCP	second-order cone programming
SOS	second-order solver
SPF	sparsity-promoting function
SPGL1	spectral projected-gradient ℓ_1-norm
TV	total-variation

List of Tables

List of Figures

ACKNOWLEDGEMENTS

The research included in this dissertation was partially supported by the Brazilian National Council for Scientific and Technological Development, CNPq, under grant 200719/2008-4, and was made possible with the assistance, patience, and support of many individuals. I would like to extend my gratitude first and foremost to my co-supervisors Dr. Stuart Bergen and Dr. Andreas Antoniou for their encouragement and dedicated guidance throughout my academic program. I would also like to thank them for giving me intellectual freedom in my work, supporting my attendance at various conferences, and demanding a high quality of work in my endeavours. I would additionally like to thank the other members of my examining committee, Dr. Jane Ye and Dr. Özgur Yilmaz, for their insightful comments that helped me improve the contents of this dissertation.

I would also like to extend my appreciation to Dr. Warren Hare from the Department of Mathematics, University of British Columbia and Dr. Alfredo Iusem from the National Institute for Pure and Applied Mathematics, IMPA, Rio de Janeiro, for sharing their expertise and for providing invaluable suggestions to this research.

Finally I would like to extend my deepest gratitude to my parents for their unconditional love and to my wife Luciana without whose love, support, and understanding I could never have completed this research.

DEDICATION

To my beloved wife Luciana and son Leo
and to the memory of
my mother Glória
my aunt Socorro
and my cousin Luíze

Chapter 1

Introduction

Compressive sensing (CS) is a recent signal processing methodology [20–22, 39] that enables the recovery of *sparse* or *compressible* signals from a very limited number of measurements, possibly contaminated by noise. The price that must be paid for compact signal reconstruction is the need of a nontrivial signal-recovery process which would involve solving a difficult optimization problem. CS is used in analog-to-digital conversion [108, 111], data compression [15, 19, 53, 65, 69, 88], medical imaging [25, 75], channel coding [5, 38] among many other applications of current interest.

The goal of the recovery process is to find the sparsest signal that is consistent with the measurements taken. Signal-recovery methods can be used to find sparse solutions by minimizing a sparsity-promoting function (SPF). Measurement consistency is achieved by ensuring that the Euclidean distance between the measurements and a linear transformation of the signal found is within a prescribed value.

In theory, the SPF would assume the form of the ℓ_0-norm function whose value is equal to the number of nonzero-valued samples of a signal. Unfortunately, the ℓ_0 norm is of little practical use because the resulting optimization problem is computationally intractable requiring a combinatorial search as specified in Theorem 1 of [80].

In contemporary signal-recovery methods [8, 9, 11, 18, 43, 68] the SPF assumes the form of a convex function such as the ℓ_1 norm or the total-variation (TV) norm. These SPFs are often employed because (1) the resulting optimization problem is convex, and (2) convex optimization problems have a fairly complete theory and can be solved efficiently. Under certain conditions, the solutions obtained by such recovery methods can be optimal, e.g., as stated in Theorem 1 of [21], Theorem 1.1 of [22], and Theorem 5 of [39].

There has been increased interest in the use of nonconvex SPFs that closely re-

semble the ℓ_0-norm function [23, 24, 26, 27, 37, 45, 46, 97, 109]. Nonconvex functions are desirable because their use can lead to shorter signal representations and reduced reconstruction error when compared with signals obtained by solving convex problems [23, 27, 37, 46]. Unfortunately, available iterative methods for the solution of nonconvex problems for CS are inefficient and are not supported by robust convergence theorems [23, 27, 46].

In this dissertation, nonconvex CS recovery problems are investigated and efficient methods and algorithms that can be used for the solutions of such problems are proposed.

1.1 Background and notation

Let $\boldsymbol{x}^0 \in \mathbb{R}^n$ denote a vector that represents the signal of interest or a transformed version of the signal in an appropriate representation. It is assumed that vector $\boldsymbol{x}^0$ is *s-sparse* in the sense that is has only s nonzero coordinates with $s < n$. The measurement process is carried out in the presence of a noise signal represented by vector $\boldsymbol{z} \in \mathbb{R}^m$ with a upper bound δ such that $||\boldsymbol{z}||_2 \leq \delta$. Acquisition of $\boldsymbol{x}^0$ is accomplished with the sensing operation

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^0 + \boldsymbol{z} \tag{1.1}$$

where $\boldsymbol{b} \in \mathbb{R}^m$ is a vector representing the measurements taken and $\boldsymbol{A}$ is an $m \times n$ sensing matrix with $m \ll n$.

CS theory revolves around random measurements with the entries of matrix $\boldsymbol{A}$ assuming independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance $1/m$ [20–22, 39]. Other measurement ensembles can be used [21] such as the Fourier ensemble where matrix $\boldsymbol{A}$ is obtained by selecting m rows at random from the $n \times n$ discrete Fourier transform (DFT) matrix and renormalizing the columns of the resulting matrix so that they have unit norm. More generally, renormalized matrix $\boldsymbol{A}$ is obtained by selecting m rows at random from an $n \times n$ orthonormal matrix. The general case of orthogonal ensembles is of practical interest because the processing can be carried out by using fast algorithms for matrix-vector products, e.g., the fast Fourier transform (FFT) algorithm for the DFT or the fast cosine transform algorithm for the discrete cosine transform (DCT), and so on.

1.1.1 Sparsity promoting functions

A measure of sparsity of vector $\boldsymbol{x} \in \mathbb{R}^n$ is given by

$$P_\epsilon(\boldsymbol{x}) = \sum_{i=1}^{n} w_i \, p_\epsilon(|x_i|) \tag{1.2}$$

where ϵ is a nonnegative regularization parameter, $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}^T$ is a vector of nonnegative weights, and $p_\epsilon(|x_i|)$ is the SPF which quantifies the magnitude of each individual coordinate of $\boldsymbol{x}$. SPFs are either convex or nonconvex functions that are carefully chosen to ensure that the minimization of $P_\epsilon(\boldsymbol{x})$ yields a sparse solution. For example, the convex SPF

$$p_\epsilon(|x_i|) = |x_i| + \epsilon \tag{1.3}$$

can be used to find sparse signals. Function $P_\epsilon(\boldsymbol{x})$ is equivalent to the ℓ_1 norm of $\boldsymbol{x}$ when $\boldsymbol{w}$ is a vector of ones and its minimizer is sparse [8, 9, 11, 18, 43, 68].

An example of a nonconvex SPF is given by

$$p_\epsilon(|x_i|) = (|x_i| + \epsilon)^p \tag{1.4}$$

where $0 < p < 1$. Here function $P_\epsilon(\boldsymbol{x})$ is called the weighted ϵ-ℓ_p^p norm of $\boldsymbol{x}$ and its minimization yields sparse solutions [23]. Plots of $p_\epsilon(|x_i|)$ for several values of parameters p and ϵ are shown in Fig. 1.1.

Another example of a nonconvex SPF is given by

$$p_\epsilon(|x_i|) = \ln(|x_i| + \epsilon) \tag{1.5}$$

It has been demonstrated in [23] that sparse solutions can be obtained by minimizing function $P_\epsilon(\boldsymbol{x})$. Plots of $p_\epsilon(|x_i|)$ for several values of parameter ϵ are shown in Fig. 1.2.

One last example of a nonconvex SPF is obtained by using the smoothly-clipped

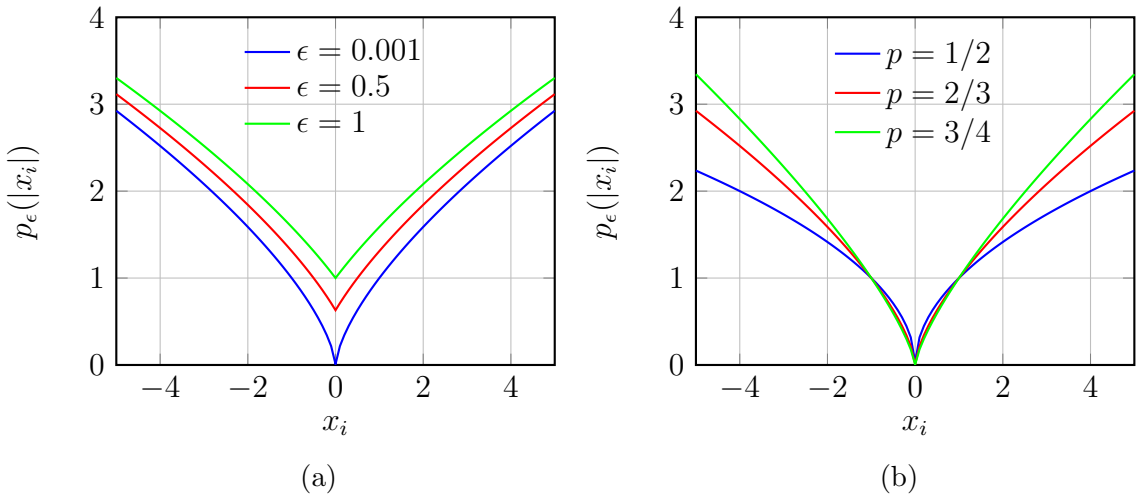


Figure 1.1: Nonconvex SPF that defines the ϵ-ℓ_p^p norm of $\boldsymbol{x}$: (a) Several values of ϵ with $p = 2/3$ and (b) Several values of p with $\epsilon = 0.001$.

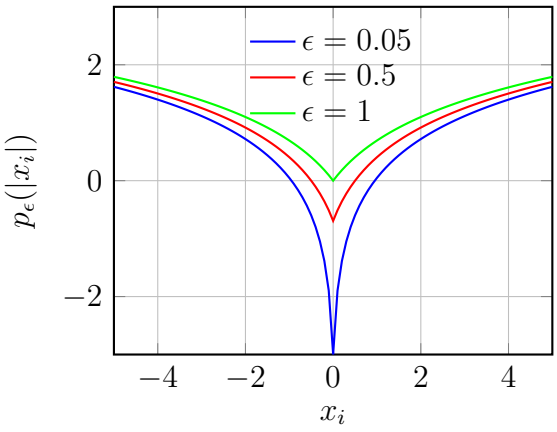


Figure 1.2: Nonconvex SPF based on the logarithm of $(|x_i| + \epsilon)$.

absolute deviation (SCAD) function given in [42]

$$p_\epsilon(|x_i|) = \begin{cases} \epsilon|x_i|, & |x_i| \leq \epsilon \\ -\left[|x_i|^2 - 2\alpha\epsilon|x_i| + \epsilon^2\right] / \left[2(\alpha - 1)\right], & \epsilon < |x_i| \leq \alpha\epsilon \\ (\alpha + 1)\epsilon^2/2, & |x_i| > \alpha\epsilon \end{cases} \tag{1.6}$$

where $\alpha > 2$. Function $P_\epsilon(\boldsymbol{x})$ has been used in [46] to find sparse signals. Plots of $p_\epsilon(|x_i|)$ for several values of parameters ϵ and α are shown in Fig. 1.3.

Hereafter we let $\mathcal{C}$ and $\mathcal{N}$ denote the classes of convex and nonconvex SPFs, respectively.

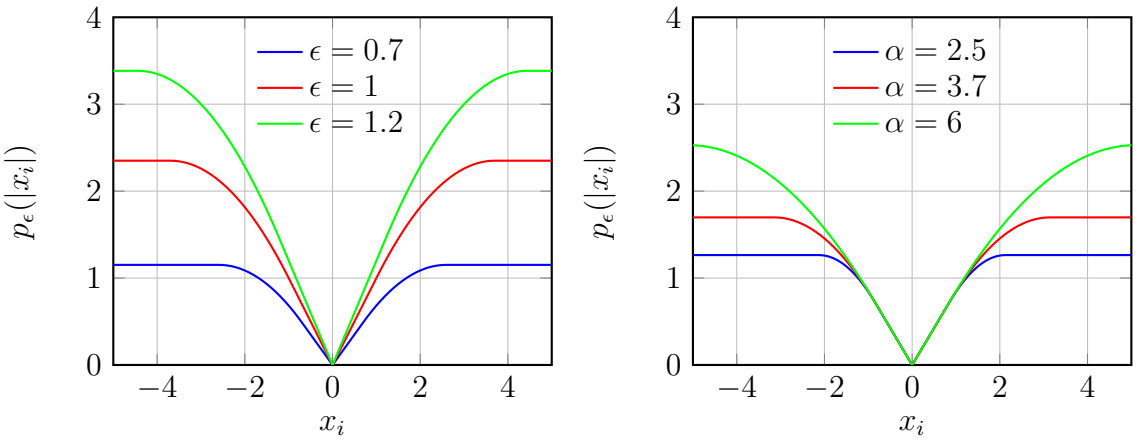


Figure 1.3: Nonconvex SPF based on the SCAD function: (a) Several values of ϵ with $\alpha = 3.7$ and (b) Several values of α with $\epsilon = 0.85$.

1.1.2 Signal recovery process

The sparse-signal recovery process can be carried out by solving closely related optimization problems. In basis pursuit (BP) methods, recovery is accomplished by solving the constrained optimization problem

$$\begin{aligned} (\mathrm{BP}_\delta) \qquad & \underset{\boldsymbol{x}}{\text{minimize}} && P_\epsilon(\boldsymbol{x}) \\ & \text{subject to:} && \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \delta \end{aligned} \tag{1.7}$$

where $\delta \geq 0$ is an estimate of the square root of the measurement noise energy. In regularized least-squares (RLS) methods, one solves the unconstrained optimization problem

$$(\mathrm{QP}_\lambda) \qquad \underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda P_\epsilon(\boldsymbol{x}) \tag{1.8}$$

where $\lambda \geq 0$ is a regularization term which controls the trade-off between measurement consistency and sparsity. On the other hand, in least absolute shrinkage and selection operator (LASSO) methods, the constrained optimization problem

$$\begin{aligned} (\mathrm{LS}_\sigma) \qquad & \underset{\boldsymbol{x}}{\text{minimize}} && \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \\ & \text{subject to:} && P_\epsilon(\boldsymbol{x}) \leq \sigma \end{aligned} \tag{1.9}$$

is solved where $\sigma \geq 0$ is a bound on the measure of sparsity of $\boldsymbol{x}$ [9, 11, 68].

In the case where $p_\epsilon(|x_i|) \in \mathcal{C}$, problems (BP_δ), (QP_λ), and (LS_σ) are closely

related because parameter λ in (QP_λ) is directly related to the Lagrange multiplier of constraint $P_\epsilon(\boldsymbol{x}) \leq \sigma$ in (LS_σ), and because it is inversely related to the Lagrange multiplier of constraint $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \delta$ in (BP_δ). Hence, these problems are equivalent for suitable choices of parameters δ, λ, and σ. Unfortunately, the relationship between these parameters is hard to compute with the exception of the case where matrix $\boldsymbol{A}$ is orthogonal [11].

1.2 State-of-the-Art Methods

Parameter δ of problem (BP_δ) is assumed to be known *a priori* in CS applications because the energy of the noise inherent in (1.1) can be readily estimated, e.g., as in physical implementations of the measurement process such as the one in [113]. BP methods are often preferred over RLS and LASSO methods for this reason. Heuristics are usually employed in RLS methods to find an *approximate* value of λ in problem (QP_λ) over which the solution found is equivalent to the solution of problem (BP_δ). Efficiencies are achieved by using *continuation procedures* [43]. In LASSO methods, all the solutions of problem (LS_σ) as a function of parameter σ are completely described. Thus, an *exact* value of σ in problem (LS_σ) over which the solution found is equivalent to the solution of problem (BP_δ) can be found. Efficiencies are achieved by using *homotopy techniques* [40, 86] or a Newton-based root finding procedure [11].

First- and second-order solvers are employed for the solution of the recovery problem. In *second-order* methods, the unconstrained problem is solved by using Newton's method or one of its variants while the constrained problem is solved by using interior-point methods. Accurate solutions are obtained by using second-order solvers (SOSs), but their use is problematic in large-scale problems as they are required to solve large systems of linear equations in computing the Newton step. In *first-order* methods, the problem is solved by using gradient methods (or subgradient methods when there is nonsmoothness) in the unconstrained case, or by using projected gradient/subgradient methods in the constrained case. First-order solvers (FOSs) are not as accurate as SOSs, but they are efficient for large-scale problems.

The recovery of realistic signals from Gaussian ensembles is problematic because matrix-vector operations cannot be carried out with fast algorithms and because FOSs and SOSs require the storage of large sensing matrices in such cases. For instance, the recovery of an image of 256×256 pixels from $25,000$ Gaussian measurements would require the storage and the manipulation of a $25,000 \times 65,536$ matrix which requires

approximately 13.6 gigabytes of memory in the case of double-precision representation [21]. In the case of orthogonal ensembles, a desirable feature of FOSs and SOSs is the capability of using matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ in matrix-vector operations only. As a result, the solver can handle realistic recovery problems because there is no need for the storage of these matrices and because matrix-vector operations can be carried out with fast algorithms. Standard SOSs such as self-dual-minimization (SeDuMi) [102] or MOSEK [1] do not possess such capability but specialized SOSs like those employed in [18] or [68] do. Recent FOSs, such as those in [8, 9, 11, 43], are also capable of handling realistic recovery problems.

1.2.1 First-order solvers

Projected gradient methods are based on the following update formula [90]

$$\boldsymbol{x}^{(k+1)} = \text{proj}_{\mathcal{X}} \left[\boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{\nabla} F(\boldsymbol{x}^{(k)})\right] \tag{1.10}$$

where $\alpha_k \geq 0$ is the step-size parameter, $F(\boldsymbol{x})$ is the objective function of the optimization problem involved, $\mathcal{X}$ denotes the problem constraint set, and $\text{proj}_{\mathcal{X}}$ denotes the projector onto this set. If we let $\boldsymbol{y}^{(k)}$ denote a point of the form

$$\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{\nabla} F(\boldsymbol{x}^{(k)})$$

then the projector in (1.10) is defined by

$$\text{proj}_{\mathcal{X}}(\boldsymbol{y}^{(k)}) = \arg \underset{\boldsymbol{x}\in\mathcal{X}}{\text{minimize}} \ \left\|\boldsymbol{x} - \boldsymbol{y}^{(k)}\right\| \tag{1.11}$$

(see p. 397 of [16]). In the case where the optimization problem involved is an unconstrained one, the update formula in (1.10) assumes the form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{\nabla} F(\boldsymbol{x}^{(k)}) \tag{1.12}$$

which corresponds to that used in gradient methods for the minimization of function $F(\boldsymbol{x})$ [90]. We let $\boldsymbol{\nabla} F(\boldsymbol{x}) = \boldsymbol{\partial} F(\boldsymbol{x})$ in (1.10) and (1.12) when $F(\boldsymbol{x})$ is a nondifferentiable function. In this case, the update formulas in (1.10) and (1.12) now correspond to a projected subgradient and subgradient methods, respectively.

The so-called *Moreau Envelope* (ME) of function $F(\boldsymbol{x})$ is given by

$$\psi_\gamma(\boldsymbol{x}) = \underset{\tilde{\boldsymbol{x}}}{\text{minimize}} \left[F(\tilde{\boldsymbol{x}}) + \frac{1}{2\gamma}||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_2^2 \right] \tag{1.13}$$

where $\gamma > 0$ (see [78]). Applying the update formula in (1.12) with $\alpha = \gamma$ and $F(\boldsymbol{x}) = \psi_\gamma(\boldsymbol{x})$ results in the closely related update formula [90]

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} - \gamma_k \boldsymbol{\nabla} \psi_{\gamma_k}(\boldsymbol{x}^{(k)}) \\ &= \text{prox}_{\gamma_k} \left[F(\boldsymbol{x}^{(k)}) \right] \end{aligned} \tag{1.14}$$

where $\text{prox}_\gamma \left[F(\boldsymbol{x}) \right]$ is the proximal-point (PP) mapping of $F(\boldsymbol{x})$ given by

$$\text{prox}_\gamma \left[F(\boldsymbol{x}) \right] = \arg \ \underset{\tilde{\boldsymbol{x}}}{\text{minimize}} \left[F(\tilde{\boldsymbol{x}}) + \frac{1}{2\gamma}||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_2^2 \right] \tag{1.15}$$

where γ and $\boldsymbol{x}$ are known as the *prox-parameter* and the *prox-center* of (1.15). If $\boldsymbol{z} \in \text{prox}_\gamma \left[F(\boldsymbol{x}) \right]$, then $\boldsymbol{z}$ is called a *PP of function* $F(\boldsymbol{x})$. The update formula in (1.14) corresponds to that used in PP methods for the minimization of function $F(\boldsymbol{x})$ (see [31, 56, 78, 89, 94] and references therein).

1.2.2 Convergence rate of specialized solvers

It is widely known that second-order methods have a much better convergence rate than first-order methods. Newton's method is capable of achieving high-accuracy solutions in a few iterations for a wide class of functions that are Lipschitz continuous, e.g., the method is efficient when applied to a function $F(\boldsymbol{x})$ with the property

$$|F(\boldsymbol{x}') - F(\boldsymbol{x})| \leq \kappa |\boldsymbol{x}' - \boldsymbol{x}| \qquad \text{for all } \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n \tag{1.16}$$

or, equivalently, with the property

$$||\boldsymbol{\nabla} F(\boldsymbol{x})||_2 \leq \kappa \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^n \tag{1.17}$$

when $F(\boldsymbol{x})$ is differentiable and with the property

$$||\boldsymbol{g}||_2 \leq \kappa \qquad \text{for all } \boldsymbol{g} \in \boldsymbol{\partial} F(\boldsymbol{x}) \text{ and } \boldsymbol{x} \in \mathbb{R}^n \tag{1.18}$$

when $F(\boldsymbol{x})$ is nondifferentiable where $\kappa \geq 0$. Function $F(\boldsymbol{x})$ is Lipschitz continuous with Lipschitz constant κ when the properties in (1.16), (1.17), or (1.18) hold true [96].

Sometimes a gradient method would require a large number of iterations to achieve reasonably accurate solutions for very simple problems [90]. It has been shown to have a worst-case convergence rate given by

$$F(\boldsymbol{x}^{(k)}) - F(\boldsymbol{x}^*) = O(1/k) \tag{1.19}$$

for recovery problems (BP_δ), (QP_λ), and (LS_σ) where $\boldsymbol{x}^*$ is the minimizer of $F(\boldsymbol{x})$ and $O(1/k)$ stands for "*is of order* $1/k$" (see Theorem 4 of [17] or Theorem 3.1 of [8]). In such a case, (1.19) implies that as $k \to \infty$

$$F(\boldsymbol{x}^{(k)}) - F(\boldsymbol{x}^*) < C\frac{1}{k}$$

where C is a constant independent of k (see Sec. 1.6 of [55]). Thus, convergence can be quite slow for such problems.

A method for speeding up the convergence of first-order methods has been described by Polyak [91]. The rate of convergence of gradient methods can be significantly improved by employing a variant of the update formula in (1.12) given by

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{\nabla} F(\boldsymbol{x}^{(k)}) + \beta_k \left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right) \tag{1.20}$$

where $0 \leq \beta_k < 1$ and $\boldsymbol{x}^{(-1)} = \boldsymbol{x}^{(0)}$. The update formula in (1.20) corresponds to the so-called *heavy-ball* or *two-step* method for the minimization of $F(\boldsymbol{x})$ [91]. The computational cost involved in the two-step method is similar to that of gradient methods because (1.20) requires only slightly more computation than (1.12).

The optimal first-order methods in [81,83] are examples of two-step methods which have received a lot of attention in the past few years because of their application in signal recovery problems (see [8, 9]). The method in [81] is essentially a two-step method for the minimization of Lipschitz differentiable functions where parameter β_k in (1.20) is chosen at each iteration such that

$$F(\boldsymbol{x}^{(k)}) - F(\boldsymbol{x}^*) = O(1/k^2) \tag{1.21}$$

(see Sec. 4.1 of [13]). The two-step method is optimal in the sense that the convergence rate in (1.21) is the highest achievable rate for the class of problems under

consideration [81]. The method in [83] uses a smooth approximation of $F(\boldsymbol{x})$ in its dual space and is applicable to the case where $F(\boldsymbol{x})$ is nondifferentiable. The approximation is shown to be Lipschitz differentiable and the optimal first-order method of [81] is applied for its minimization.

1.2.3 First- and second-order solvers in nonconvex problems

The solution techniques discussed so far are directly applicable to the problems in (1.7), (1.8), and (1.9) in the case where $p_\epsilon(|x_i|) \in \mathcal{C}$. In fact, several specialized SOSs and FOSs based on such solution techniques have been perfected over the past few years [8,10,18,43,68]. These solvers are driven by advances in the theory and methods for the solution of convex programming problems. In the case where $p_\epsilon(|x_i|) \in \mathcal{N}$, signal-recovery methods employ either an indirect or direct approach to the solution of the nonconvex optimization problem involved.

In the indirect approach, approximation is employed. Nonconvex optimization problems are relaxed into a sequence of convex subproblems which can be solved efficiently using specialized FOSs or SOSs. The problem of minimizing a nonconvex objective function $F(\boldsymbol{x})$ over a convex set $\mathcal{X}$ is approached by (1) finding an approximation of the solution $\boldsymbol{x}^{(0)} \in \mathcal{X}$ that can be used as an initial point, and by (2) using the update formula

$$\boldsymbol{x}^{(k+1)} = \arg \underset{\boldsymbol{x}\in\mathcal{X}}{\text{minimize}}\ \widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{1.22}$$

where convex function $\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ denotes an approximation of $F(\boldsymbol{x})$ at the current point $\boldsymbol{x}^{(k)}$. As a result of the approximation, the next point $\boldsymbol{x}^{(k+1)}$ is obtained as the solution of a convex problem. The update formula in (1.22) defines what we call a sequential convex formulation (SCF) method for this reason. The method is applicable when the sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.22) converges to a solution of the original nonconvex problem.

In the direct approach, nonconvex objective functions with convex and differentiable MEs and single-valued PP mappings are employed. These unusually rich properties can be found in a large range of functions of interest in variational analysis such as lower-C^2 and prox-regular functions (see [89, 96] and references therein). A function $F(\boldsymbol{x})$ is said to be *lower-C^2 on* $\mathbb{R}^n$ if there exists a constant $\rho > 0$ such that $F(\boldsymbol{x})$ can be written as

$$F(\boldsymbol{x}) = h(\boldsymbol{x}) - \frac{1}{2}\rho||\boldsymbol{x}||_2^2 \tag{1.23}$$

where $h(\boldsymbol{x})$ is a convex function (see Theorem 10.33 of [96]). On the other hand, function $F(\boldsymbol{x})$ is said to be *prox-regular at* $\bar{\boldsymbol{x}}$ for $\bar{\boldsymbol{v}}$ where $\bar{\boldsymbol{v}} \in \boldsymbol{\partial} F(\bar{\boldsymbol{x}})$, if there exist constants $r > 0$ and $\varepsilon > 0$ such that

$$F(\tilde{\boldsymbol{x}}) \geq F(\boldsymbol{x}) + \boldsymbol{v}^T(\tilde{\boldsymbol{x}} - \boldsymbol{x}) - \frac{r}{2}||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_2^2 \tag{1.24}$$

whenever $||\tilde{\boldsymbol{x}}-\bar{\boldsymbol{x}}||_2 < \varepsilon$, $||\boldsymbol{x}-\bar{\boldsymbol{x}}||_2 < \varepsilon$, and $||F(\boldsymbol{x})-F(\bar{\boldsymbol{x}})||_2 < \varepsilon$ with $\tilde{\boldsymbol{x}} \neq \boldsymbol{x}$, while $||\boldsymbol{v}-\bar{\boldsymbol{v}}||_2 < \varepsilon$ where $\boldsymbol{v} \in \boldsymbol{\partial} F(\boldsymbol{x})$ (see Definition 1.1 of [89]). Because of the aforementioned properties, results pertaining to the convergence of sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.14) for the case where $F(\boldsymbol{x})$ is convex, such as those in [94], can be extended to a nonconvex setting (see [30, 63, 87]). Thus, PP methods are applicable to the solution of the nonconvex optimization problem involved.

The remainder of this section describes representative RLS, LASSO, and BP methods in both convex and nonconvex recovery settings.

1.2.4 RLS Methods

RLS methods date back to the work of Tikhonov [107] where problem (QP_λ) has been proposed for finding an approximate solution of $\boldsymbol{A}x = \boldsymbol{b}$ when the $m \times n$ matrix $\boldsymbol{A}$ is ill-conditioned or singular. In such ill-posed problems, the solution

$$\boldsymbol{x}_{\text{LS}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$$

of the least-squares problem given by

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 \tag{1.25}$$

is a poor approximation to $\boldsymbol{x}$ [84]. In RLS methods, meaningful solution estimates can be obtained for ill-posed problems. This is achieved by employing regularization techniques where the problem in (1.25) is regularized by the addition of a term of the form $\lambda P_\epsilon(\boldsymbol{x})$ with $\lambda > 0$. In the so-called *Tikhonov regularization*, a good approximate can be obtained by solving problem (QP_λ) with $p_\epsilon(|x_i|) = |x_i|^2$ for overdetermined problems with $m > n$.

The use of RLS methods for obtaining sparse solutions to ill-posed problems dates back to the work of Santosa and Symes [98] where problem (QP_λ) with $p_\epsilon(|x_i|) = |x_i|$ was used for finding the inverse of bandlimited reflection seismograms. The use of

RLS methods for estimating linear models in statistics has been proposed by Fan and Li in [42]. In the case where the columns of matrix $\boldsymbol{A}$ are orthonormal, the particular form assumed by the SPF is directly related to the *unbiasedness*, *sparsity*, and *continuity* of the solution $\boldsymbol{x}^*$ of problem (QP_λ). For instance, unbiasedness is achieved when $p'(|x_i|) = 0$ for sufficiently large values of $|x_i|$ because $\boldsymbol{x}^*$ is consistent with vector $\boldsymbol{y} = \boldsymbol{A}^T\boldsymbol{b}$ with high probability. On the other hand, when the minimum of function $g(|x_i|) = |x_i| + p'(|x_i|)$ is positive a sparse solution is obtained. Finally, when the minimum of $g(|x_i|)$ occurs at zero, $\boldsymbol{x}^*$ is continuous with respect to $\boldsymbol{y}$ in the sense that small perturbations to $\boldsymbol{y}$ yield small perturbations in $\boldsymbol{x}^*$ (see Sec. 2 of [42] for details).

Specialized RLS methods have been proposed for solving signal recovery problems when $p_\epsilon(|x_i|) \in \mathcal{C}$ [8, 43, 68]. Because the objective function of problem (QP_λ) is not differentiable, most methods approach a solution indirectly by recasting it as a convex quadratic programming (QP) problem with linear inequality constraints, namely,

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{u}}{\text{minimize}} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda \sum_{i=1}^{n} p_\epsilon(u_i) \\ \text{subject to:} \quad & -u_i \leq x_i \leq u_i,\ i = 1,\ \cdots,\ n \end{aligned}$$

In the ℓ_1-LS method, the above problem is solved by employing a customized primal interior-point method [68]. Inequality constraints are removed by adding a logarithmic barrier term to the objective function. The resulting unconstrained problem is solved with Newton's method, and a preconditioned conjugate gradient (PCG) algorithm [67] is used for solving the linear system of equations associated with computing the search direction. In addition, the SOS can handle large-scale recovery problems because the PCG algorithm utilizes matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ in matrix-vector operations only. Experiments carried out with the ℓ_1-LS method suggest the use of $\lambda = 0.1||2\boldsymbol{A}^T\boldsymbol{b}||_\infty$ for recovering signals in CS applications [68].

Another RLS method that approaches a solution of problem (QP_λ) by recasting it as a convex QP problem is the gradient projection for sparse reconstruction (GPSR) method [43]. This is achieved by splitting variable $\boldsymbol{x}$ of problem (QP_λ) into its positive $\boldsymbol{u}$ and negative $\boldsymbol{v}$ parts for $\boldsymbol{x} = \boldsymbol{u} - \boldsymbol{v}$ with $\boldsymbol{u} \geq \boldsymbol{0}$ and $\boldsymbol{v} \geq \boldsymbol{0}$. Hence, problem ($\text{QP}_\lambda$) is recast as a bound-constrained quadratic programming (BCQP) problem given by

$$\underset{\boldsymbol{z}\in\mathcal{X}}{\text{minimize}}\ F(\boldsymbol{z}) \triangleq \left\{\boldsymbol{\lambda} + \begin{bmatrix} -\boldsymbol{A}^T\boldsymbol{b} & \boldsymbol{A}^T\boldsymbol{b} \end{bmatrix}\right\}\boldsymbol{z} + \frac{1}{2}\boldsymbol{z}^T \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & -\boldsymbol{A}^T\boldsymbol{A} \\ -\boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{A}^T\boldsymbol{A} \end{bmatrix}\boldsymbol{z}$$

where $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{u} & \boldsymbol{v} \end{bmatrix}^T$ and $\mathcal{X} = \{\boldsymbol{z} \in \mathbb{R}^{2n} : z_i \geq 0,\ i = 1, \ldots, 2n\}$. The GPSR method uses the update formula in (1.10) for minimizing $F(\boldsymbol{z})$. Each point in the iteration sequence can be efficiently computed because the computation of the orthogonal projector has a simple analytical solution and the computation of the gradient requires one multiplication each by matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$. Hence, the processing can be carried out by using fast algorithms for matrix- vector products in the case of orthogonal ensembles. Different techniques such as the *backtracking* or the *Barzilai-Borwen* methods are employed for choosing step-size α_k to speed up the convergence of the iteration sequence in (1.10) [43]. As in the ℓ_1-LS method, the heuristic $\lambda = 0.1||2\boldsymbol{A}^T\boldsymbol{b}||_\infty$ is used for recovering signals in CS applications.

Finally, another recent RLS method for the case of $p_\epsilon(|x_i|) \in \mathcal{C}$ is the so-called fast iterative shrinkage-thresholding algorithm (FISTA) [8]. The FISTA method approaches a solution to problem (QP_λ) directly by employing a PP method. Sequence $\left\{\boldsymbol{a}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.14) can be computed as [8]

$$\begin{aligned} \boldsymbol{a}^{(k+1)} &= \arg \underset{\boldsymbol{x}}{\text{minimize}} \left[\lambda P_\epsilon(\boldsymbol{x}) + \frac{1}{2}\gamma_k \left\| \boldsymbol{x} - \left(\boldsymbol{a}^{(k)} - \frac{1}{\gamma_k}\boldsymbol{\nabla}||\boldsymbol{A}\boldsymbol{a}^{(k)} - \boldsymbol{b}||_2^2\right)\right\|_2^2\right] \\ &= \text{prox}_{\gamma_k}\left[\lambda P_\epsilon\left(\boldsymbol{a}^{(k)} - \frac{1}{\gamma_k}\boldsymbol{\nabla}||\boldsymbol{A}\boldsymbol{a}^{(k)} - \boldsymbol{b}||_2^2\right)\right] \end{aligned} \tag{1.26}$$

where $\gamma_0 = \gamma_1 = \cdots = \gamma_k$ are equal to the inverse of the Lipschitz constant of $\boldsymbol{\nabla}(||\boldsymbol{A}\boldsymbol{a}^{(k)} - \boldsymbol{b}||_2^2)$. In the FOS, a solution to problem (QP_λ) can be found efficiently because computation of the PP in (1.26) has a simple analytical solution given by the so-called *thresholding-shrinkage* operation for $p_\epsilon(|x_i|) = |x_i|$. In addition, the computation of the gradient in (1.26) involves only matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$. Hence, the processing can be carried out by using fast algorithms for matrix-vector products in the case of orthogonal ensembles. Acceleration of the convergence of sequence $\left\{\boldsymbol{a}^{(k)}\right\}$ is achieved by using the update formula in (1.20) with

$$\beta_k = \frac{t_k - 1}{t_{k+1}} \tag{1.27}$$

where

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \tag{1.28}$$

and $t_0 = 1$. Parameter β_k in (1.27) is the same as the one used in the optimal first-order method in [81]. Thus, the rate of convergence of the FISTA method is given by

(1.21).

Recently, a family of RLS methods has been proposed for the solution of problem (QP_λ) in the case where $p_\epsilon(|x_i|) \in \mathcal{N}$ [46]. In these methods, sparsity is promoted with function $P_\epsilon(\boldsymbol{x})$ as given in (1.2) where $\boldsymbol{w}$ is a vector of ones and the nonconvex SPF can be written as

$$p_\epsilon(|x_i|) = g(|x_i|) - h_\epsilon(|x_i|) \tag{1.29}$$

where $g(|x_i|)$ and $h_\epsilon(|x_i|)$ are convex functions [46]. The difference-of-two-convex-functions (DC) programming approach [60] is employed for solving the nonconvex problem. The solution method is based on the update formula in (1.22) where an approximation of the objective function $F(\boldsymbol{x})$ of the problem in (1.8) is constructed at each step k as

$$\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 + \widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x})$$

where convex function $\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x})$ is an approximation of $P_\epsilon(\boldsymbol{x})$ at the solution point $\boldsymbol{x}^{(k)}$. Using the DC decomposition in (1.29), function $\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x})$ can be written as

$$\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i^{(k)} \, |x_i|$$

which is equivalent to the weighted ℓ_1 norm of $\boldsymbol{x}$. The solution point $\boldsymbol{x}^{(k)}$ is used for the computation of the weight vector $\boldsymbol{w}^{(k)}$. When $\boldsymbol{w}^{(k)}$ is appropriately chosen, the sequence of points $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.22) converges to a minimizer of the original nonconvex problem [46]. Each subproblem in (1.22) is convex and solved with the GPSR method [43]. A DC method where a solution of problem (QP_λ) is obtained for $p_\epsilon(|x_i|)$ given by (1.4), (1.5), and (1.6) will hereafter be called $\text{DC}_{\ell_p^p}$, DC_{ln}, and DC_{SCAD}, respectively.

The properties of the solutions of problem (QP_λ) with respect to parameter λ are addressed in [68]. When $p_\epsilon(|x_i|) = |x_i|$, the solution of problem (QP_λ) for $\lambda \to 0$ has the minimum ℓ_1 norm among all points that satisfy equation $\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{0}$. Moreover, the solution of (QP_λ) tends to the zero vector in the limiting point $\lambda \to \infty$. In such a case, however, it has been observed that convergence occurs for a finite value of $\lambda \geq ||2\boldsymbol{A}^T\boldsymbol{b}||$ [68]. The convergence rate of the iteration sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.10) or (1.12) slows for decreasing values of λ [43]. In other words, the computational cost of FOSs may increase considerably as $\lambda \to 0$. Because a small λ is usually required in CS applications, in recent RLS methods reduced computational

cost is achieved by using continuation procedures [8, 43, 68]. In such procedures, several (QP_λ) problems are solved in sequence for decreasing values of λ starting with $\lambda = ||2\boldsymbol{A}^T\boldsymbol{b}||_\infty$. The solution point of the previous problem is used as the initial point for the next one. Such procedures reduce the computation required for the recovery process because the convergence rate of the FOS is improved when an appropriate initialization is employed [43].

1.2.5 LASSO Methods

LASSO methods originate from the work of Tibishiran [106] where problem (LS_σ) has been proposed for estimating linear models in statistics. The use of bound σ on function $P_\epsilon(\boldsymbol{x})$ forces zero components in the minimizing solution for small values of σ. Problem (LS_σ) is usually solved by standard SOSs because it can be recast as a QP problem in the case where $p_\epsilon(|x_i|) = |x_i|$ (see Sec. 6 of [106]). LASSO methods are preferred over RLS methods in CS applications because efficient methods exist for representing all solutions of problem (LS_σ) as a function of parameter σ [11, 40, 86]. Hence, the exact value of σ in problem (LS_σ) that solves problem (BP_δ) can be found with a minimal amount of computation.

The use of LASSO methods for the recovery of sparse signals in the case where $p_\epsilon(|x_i|) \in \mathcal{C}$ has been proposed in [11]. In the so-called spectral projected-gradient ℓ_1-norm (SPGL1) method, problem (BP_δ) is posed as the problem of finding the root of a single-variable nonlinear equation. At each iteration, an estimate of that variable is used to define a convex optimization problem whose solution yields derivative information that can be used in a Newton-based root finding algorithm [11]. The convex optimization problem is solved with a specialized FOS and the processing can be carried out by using fast algorithms for matrix-vector products in the case of orthogonal ensembles. The solver is based on the update formula in (1.10) and it entails the computation of the orthogonal projector onto set $||\boldsymbol{x}||_1 \leq \sigma$. Such a projector can be efficiently computed as described in Sec. 4.2 of [11].

1.2.6 BP Methods

BP methods originate from the work of Chen et al. [28] where a problem of similar form to (BP_δ) has been proposed for obtaining optimal signal representations in *overcomplete* dictionaries. The signal representation found is optimal in the sense that it has the smallest ℓ_1 norm among all representations in the dictionary, i.e., a

solution of (BP_δ) is found for $p_\epsilon(|x_i|) = |x_i|$. BP methods are preferred for carrying out the recovery process because parameter δ is known in advance and, therefore, a direct solution to problem (BP_δ) can be found.

Several BP methods have been proposed for the case where $p_\epsilon(|x_i|) \in \mathcal{C}$. In the ℓ_1-Magic method [18], a specialized SOS has been proposed for the solution of problem (BP_δ). In the case of a noiseless measurement process, i.e., when $\delta = 0$ in (1.1), the problem (BP_δ) reduces to the following optimization problem

$$\begin{aligned} \text{(BP)} \qquad & \underset{\boldsymbol{x}}{\text{minimize}} \qquad P_\epsilon(\boldsymbol{x}) \\ & \text{subject to:} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned} \tag{1.30}$$

and is solved with a primal-dual interior-point solver because it can be recast as a linear programming (LP) problem. When $\delta > 0$, the problem (BP_δ) is recast as a second-order cone programming (SOCP) problem and is solved with a log-barrier interior-point solver. The processing can be carried out by using fast algorithms for matrix-vector products because the SOS employs a conjugate gradient solver to find an approximate solution to the systems of linear equations involved in computing the Newton step.

Another BP method for the case where $p_\epsilon(|x_i|) \in \mathcal{C}$ has recently been proposed in [9]. When function $P_\epsilon(\boldsymbol{x})$ is equivalent to the weighted ℓ_1 norm of $\boldsymbol{x}$, it can be written as

$$P_\epsilon(\boldsymbol{x}) = \underset{\boldsymbol{u} \in \mathcal{Q}}{\text{maximize}} \ \langle \boldsymbol{u}, \boldsymbol{W}\boldsymbol{x} \rangle \tag{1.31}$$

where $\boldsymbol{W} = \text{diag}(w_1,\ w_2,\ \ldots,\ w_n)$ is a matrix with diagonal entries defined by the nonnegative weights in (1.2) and $\mathcal{Q} = \{\boldsymbol{u} \in \mathbb{R}^n : ||\boldsymbol{u}||_\infty \leq 1\}$ is the dual feasible set of function $P_\epsilon(\boldsymbol{x})$ [9]. In the so-called NESTA method, the smoothing technique proposed in [83] is employed to obtain a Lipschitz continuous approximation of function $P_\epsilon(\boldsymbol{x})$. In such a case, an approximation is given by [9]

$$\widehat{P}_{\mu,\epsilon}(\boldsymbol{x}) = \underset{\boldsymbol{u} \in \mathcal{Q}}{\text{maximize}} \ \left(\langle \boldsymbol{u}, \boldsymbol{W}\boldsymbol{x} \rangle - \frac{\mu}{2} ||\boldsymbol{u}||_2^2 \right) \tag{1.32}$$

where $\mu > 0$ is the smoothness parameter. The solution method boils down to finding a solution to the saddle-point problem

$$\underset{\boldsymbol{x} \in \mathcal{K}}{\text{minimize}} \ \widehat{P}_{\mu,\epsilon}(\boldsymbol{x}) \tag{1.33}$$

where $\mathcal{K} = \{\boldsymbol{x} \in \mathbb{R}^n : ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2 \leq \delta\}$. The solver for the above problem employs the optimal first-order method in [81] since the smooth approximation $\widehat{P}_{\mu,\epsilon}(\boldsymbol{x})$ is a Lipschitz continuous function. In addition, matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ can be used in matrix-vector operations. Thus, the FOS is efficient because (1) the processing can be carried out by using fast algorithms for matrix- vector products in the case of orthogonal ensembles, and (2) the convergence rate is the best achievable rate for the problem under consideration, just as in (1.21).

The so-called iteratively reweighted least squares (IRWLS) method has recently been proposed as a method for the recovery of sparse signals in CS in the case where $p_\epsilon(|x_i|) \in \mathcal{N}$ [27]. The IRWLS method has a long history in the literature of mathematical optimization, which made its first appearance in its current form in the doctoral thesis of Lawson [72] (see Sec. 1 of [37] for details). The use of IRWLS methods for the recovery of sparse signals dates back to the work of Gorodnitsky et al. [51]. In the case of a noiseless measurement process, function $P_\epsilon(\boldsymbol{x})$ as given in (1.2) is used in the recovery process where $\boldsymbol{w}$ is a vector of ones and the nonconvex SPF is of the form $p_\epsilon(|x_i|) = |x_i|^p$ for $0 < p < 1$ [27]. In such a case, the nonconvex function $P_\epsilon(\boldsymbol{x})$ is equivalent to the ℓ_p^p norm of $\boldsymbol{x}$. The solution method is based on the update formula in (1.22) where an approximation of the objective function of the problem in (1.30) is constructed at each step k as [27]

$$\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i^{(k)} \, |x_i|^2$$

where $w_i^{(k)} = |x_i^{(k)}|^{p-2}$. Thus, convex function $\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x})$ is equivalent to computing the weighted ℓ_2 norm of $\boldsymbol{x}$. The convergence properties of the IRWLS method have been addressed in [37]. It is shown that when certain conditions are imposed on matrix $\boldsymbol{A}$, the sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.22) converges to a local minimizer of the original nonconvex optimization problem. In addition, it is shown that the convergence rate of such a sequence is superlinear and it approaches a quadratic rate when p approaches 0 [37]. Each resulting subproblem in (1.22) is a convex one with the analytical solution [27]

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{W}^{(k)}\boldsymbol{A}^T\left(\boldsymbol{A}\boldsymbol{W}^{(k)}\boldsymbol{A}^T\right)^{-1}\boldsymbol{b} \tag{1.34}$$

where $\boldsymbol{W}^{(k)} = \text{diag}(w_1^{(k)}, \, w_2^{(k)}, \, \ldots, \, w_n^{(k)})$. The use of (1.34) is problematic for large-scale problems because it requires the solution of a large system of linear equations. To the author's knowledge, there are no specialized FOSs or SOSs for weighted ℓ_2

norm problems capable of using fast algorithms for matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$.

Finally, another BP method in the case where $p_\epsilon(|x_i|) \in \mathcal{N}$ is the so-called iterative reweighted ℓ_1 (IRWL1) method [23]. In this method, sparsity is promoted with function $P_\epsilon(\boldsymbol{x})$ as given in (1.2) where $\boldsymbol{w}$ is a vector of ones and the nonconvex SPF is of the form

$$p_\epsilon(|x_i|) = \log(|x_i| + \epsilon)$$

The solution method is based on the update formula in (1.22) where an approximation $\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x})$ of the objective function of the resulting nonconvex problem in (1.7) is constructed at each step k as [23]

$$\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i^{(k)} \, |x_i| \tag{1.35}$$

with $w_i^{(k)} = 1/(|x_i^{(k)}| + \epsilon)$. Thus, convex function $\widehat{P}_{\boldsymbol{x}^{(k)},\epsilon}(\boldsymbol{x})$ is equivalent to computing the weighted ℓ_1 norm of $\boldsymbol{x}$. Each subproblem in (1.22) is a convex one and it is solved with the ℓ_1-Magic method [18]. The solution method belongs to the class of so-called majorization-minimization (MM) methods (see [62, 70, 71] for details).

1.3 Experimental Protocol

Signal-recovery methods are evaluated in terms of their capability of recovering sparse signals in a wide range of test problems. Hereafter small-, medium-, large-, and very-large-scale loosely apply to test problems where $n < 2^{12}$, $2^{12} < n < 2^{16}$, $2^{16} < n < 2^{18}$, and $n > 2^{18}$, respectively.

Suitable metrics can be used to measure the reconstruction performance (RP), the measurement consistency (MC) of the recovered signals, and the computational cost (CC) of signal reconstruction. These metrics can be estimated by carrying out the recovery process over diverse sets of measurements several times in numerical computing environments such as MATLAB. Diversity can be achieved when (1) m measurements taken are based on s-sparse signals of size n and different values of s and n are used and (2) sparse signals are generated at random. A widely employed RP metric is given in terms of the ℓ_∞ reconstruction error defined as

$$e_\infty = ||\boldsymbol{x}^* - \boldsymbol{x}^0||_{\ell_\infty}$$

where $\boldsymbol{x}^0$ is the known signal of interest, and $\boldsymbol{x}^*$ is the signal found from problem (BP_δ), (LS_σ), or (QP_λ). Another widely employed RP metric is defined in terms of the probability of perfect recovery (PPR). Perfect recovery is declared when the signal recovered is sufficiently close to the known signal, i.e., when the inequality

$$||\boldsymbol{x}^* - \boldsymbol{x}^0||_{\ell_\infty} \leq \nu \tag{1.36}$$

holds true where $\nu > 0$ is a small constant. Because there exists a directly proportional relationship between the values of m and s when perfect recovery is achieved for a given value of n (see discussion on p. 739 of [76]), RP can also be measured by estimating the minimum required fraction (MRF), m/s, for perfectly recovering signals of size n. The use of signal-recovery methods that achieve small MRFs is desirable from a practical point of view because the number of measurements required to represent sparse signals can be reduced.

An MC metric is given in terms of the difference between the Euclidean distance $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ and the estimate of the square root of the measurement noise energy δ. If we suppose that the recovery process is carried out t times, a data set is obtained by collecting t recovered signals and by arranging the values of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ in ascending order of magnitude. The deviation between the values of this data set and that of δ can be illustrated by constructing the box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ as shown in Fig. 1.4. The box

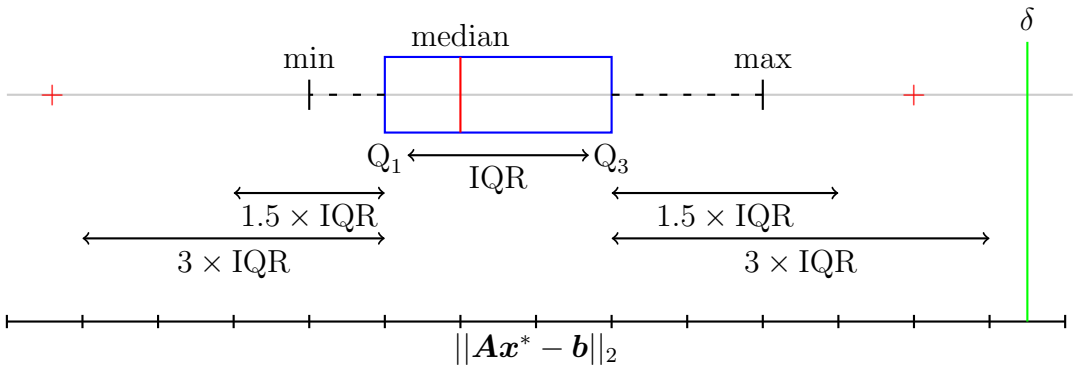


Figure 1.4: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ illustrated.

plot is a widely used tool in exploratory data analysis and the five-number summary of the data set in terms of the minimum and maximum observations, the median, and the lower and upper quartiles are usually employed (see [77] and references therein). When t is odd, the median, and the lower Q_1 and upper Q_3 quartiles of the data set are given, respectively, by the $\frac{1}{2}(t+1)$th, $\frac{1}{4}(t+1)$th, and $\frac{3}{4}(t+1)$th values of the

rearranged data set. The interquartile range is given by IQR $= \mathrm{Q}_3 - \mathrm{Q}_1$. Values larger than $\mathrm{Q}_3 + 1.5 \times \mathrm{IQR}$ or smaller than $\mathrm{Q}_1 - 1.5 \times \mathrm{IQR}$ are considered *outliers*. These limits are represented by the red "+" signs.

A CC metric is given in terms of the average CPU time required to carry out the recovery process in a specified number of trials. CPU time can be obtained by using MATLAB built-in stopwatch timers, e.g., the *tic-toc* command. Another CC metric is given in terms of the average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$. The number of such operations can be obtained by incrementing counters when the matrices involved are used during the recovery process. The CC entailed by matrix-vector operations is usually the dominant cost involved in the recovery of large signals.

Numerical simulations are conducted by evaluating the aforementioned metrics for the recovery of s-sparse signals where the s nonzero values are chosen randomly from a zero-mean Gaussian distribution of unit variance. By setting the nonzero values at random, the results are not forced to obey any particular pattern. Alternatively, the s nonzero values of the known signal of interest can be generated as [9]

$$x_i^0 = \eta_i \, 10^{\zeta_i \kappa} \tag{1.37}$$

where η_i denotes a random sign, i.e., $\eta_i = \pm 1$ with probability $1/2$, ζ_i is uniformly distributed in $[0, 1]$, and parameter κ quantifies the dynamic range (DR) of signal $\boldsymbol{x}^0$. A signal with DR of d dB is obtained in (1.37) by letting $\kappa = d/20$. Such signals are recovered from Gaussian or orthogonal ensembles. Renormalized sensing matrices obtained from Gaussian or orthonormal matrices are employed for generating these ensembles. In the signal recovery for noisy signals, measurement vector $\boldsymbol{b}$ is obtained as in (1.1) under the presence of a Gaussian noise vector $\boldsymbol{z}$ assuming a standard deviation $\sigma_{\boldsymbol{z}}$. In the case of noiseless signals, we let the standard deviation of vector $\boldsymbol{z}$ be zero, in which case (1.1) reduces to $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^0$.

1.4 Original Contributions

As detailed in Sec. 1.2, two major classes of signal-recovery methods can be identified. Methods in which $p_\epsilon(|x_i|) \in \mathcal{C}$ [8, 9, 11, 18, 43, 68] and methods in which $p_\epsilon(|x_i|) \in \mathcal{N}$ [23, 27, 46]. The former are based on specialized efficient solvers but deliver inferior RP metrics relative to the latter when only a limited number of measurements is available

[23, 24, 26, 27, 37, 46, 109]. Methods of the second class lead to reduced reconstruction error but they are either inefficient or not supported by robust convergence theorems. For instance, the family of RLS methods in [46] can perform the recovery process with several nonconvex SPFs and generates a sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ in (1.22) that converges to a local minimizer of the nonconvex optimization problem involved. However, these methods are not efficient because they entail the solution of a sequence of (QP_λ) problems for decreasing values of λ. The IRWLS method [27] is supported by a detailed analysis pertaining to the convergence of sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ in (1.22) to a local minimizer but it cannot be used to recover noisy signals. In addition, it lacks a specialized solver for each convex subproblem in (1.22) that can exploit fast algorithms for matrix-vector operations. On the other hand, the IRWL1 method [23] is capable of carrying out the recovery process under realistic circumstances but it lacks an analysis pertaining to the convergence of sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ in (1.22). In this dissertation, new efficient signal-recovery methods are proposed that outperform several state-of-the-art methods.

The proposed methods fall into two categories, namely, SCF based methods and PP based methods. The methods are supported by robust convergence theorems and they are based on efficient solvers such as SOSs suitable for recovering signals from Gaussian ensembles and specialized FOSs capable of handling large-scale recovery problems for orthogonal ensembles. The relation between the proposed and competing methods with respect to the type of SPF, recovery problem, and solver employed is shown in Figure 1.5. In this figure, the proposed and competing methods are highlighted in yellow and red, respectively.

The proposed methods are of practical use in CS and more generally in the field of signal processing because numerical results demonstrate that (1) they lead to shorter more compact signal representations than representations obtained with state-of-the-art methods while requiring a comparable amount of computation, and (2) they can solve hard realistic recovery problems of large DR and scale. The proposed methods are described in detail below.

In Chapter 2, we propose two closely related SCF methods that are applicable for the recovery of sparse signals from Gaussian ensembles. Sparsity is promoted by using the SCAD function which is known to satisfy certain conditions of unbiasedness, sparsity, and continuity in linear regression analysis described in [42]. Convex approximations of the SCAD function such as the quadratic approximation (QA) and the piecewise-linear approximation (PLA) are employed to render computation of the

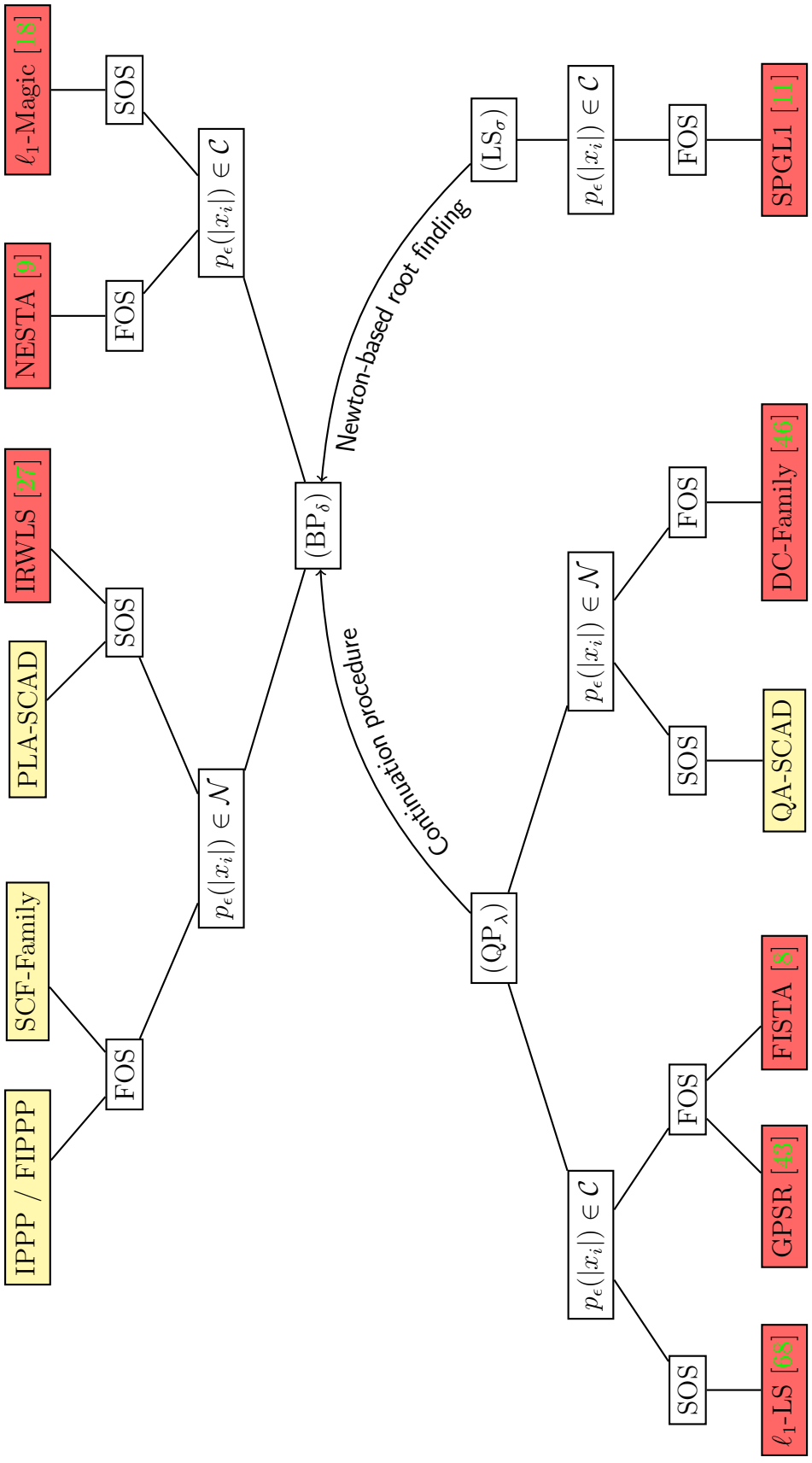


Figure 1.5: Proposed and competing signal-recovery methods in CS

minimizer tractable. In the first method, a solution of problem (QP_λ) is approached by employing the QA of the SCAD function. Convex subproblems are solved by using an SOS where the Newton step can be computed efficiently. A target value of the regularization term of the recovery problem is approached efficiently by using a continuation procedure. In the second method, a solution to problem (BP_δ) is approached by employing the PLA of the SCAD function. Convex subproblems are reformulated as SOCP problems and are solved efficiently by using standard SOSs such as SeDuMi. Numerical simulations demonstrate that the proposed methods achieve superior RP metrics in terms of increased PPRs and reduced MRFs for perfect recovery when compared with corresponding competing methods.

In Chapter 3, a new family of SCF methods is proposed, which are suitable for large-scale recovery problems. Sparsity is promoted with a fairly general class of nonconvex SPFs that include widely used SPFs as special cases. A convex approximation for the SPF such as the PLA is employed to render computation of the minimizer tractable. In the new family of SCF methods, subproblems are formulated as weighted ℓ_1-norm minimization problems while an efficient FOS suitable for the recovery of large signals from Gaussian or orthogonal ensembles is employed. The sequence of solution points is shown to be a monotonically decreasing sequence of values of the objective function and, consequently, converges to a sparse minimizer. Simulation results demonstrate that the new methods are robust, lead to fast convergence, and yield solutions that are superior to those achieved with some competing state-of-the-art methods.

In Chapter 4, a new PP based method that solves very-large-scale nonconvex optimization problems is proposed. Sparse-signal recovery is carried out by minimizing the sum of a nonconvex SPF and the indicator function of a convex set. The objective function obtained in this way exhibits unusually rich properties from an optimization perspective. A PP method is used for minimizing the objective function and a continuation procedure is employed so that a minimum can be efficiently obtained. When the iteration sequence involved is computed approximately, the method can be applied by iteratively performing two fundamental operations, namely, computation of the PP of the SPF and projection of the PP onto a convex set. The first operation is performed either analytically or numerically by using a fast iterative method while the second operation is performed by computing a sequence of closed-form projectors. The sequence of points associated with the iterative computation is shown to converge to a minimizer of the problem at hand and a two-step method with opti-

mal convergence rate is employed for accelerated convergence. Simulations carried out with the proposed method show that very-large signals can be recovered, typically in the range of a million samples, and that the solutions obtained are superior to those obtained with competing state-of-the-art methods while requiring a comparable amount of computation.

Finally, in Chapter 5 we draw conclusions and make recommendations for future research.

Chapter 2

Sequential Convex Formulation Methods Based on the Smoothly Clipped Absolute Deviation Function

2.1 Introduction

The conditions on a sparsity-promoting function (SPF) for unbiasedness, sparsity, and continuity of the solution $\boldsymbol{x}^*$ of problem (QP_λ) in (1.8) (see discussion on Sec. 1.2.4) are satisfied when $p_\epsilon(|x_i|)$ is equivalent to the smoothly-clipped absolute deviation (SCAD) function [42]. Here we are searching for SPFs that yield unbiased, continuous, and sparse solutions.

We propose to utilize the SCAD function in signal recovery problems because (1) the SPF in (1.6) satisfies the conditions for unbiased, continuous, and sparse solutions simultaneously unlike most widely used SPFs of class $\mathcal{C}$ and $\mathcal{N}$ such as those in (1.3) and (1.4), and (2) regularized least-squares (RLS) methods based on the SCAD function have the so-called "oracle property" when parameter ϵ in (1.6) is appropriately chosen [42]. A recovery method is said to have the oracle property if zero-valued coordinates of the signal of interest $\boldsymbol{x}^0$ are recovered as 0 with probability tending to 1, and the nonzero-valued coordinates are recovered with the same efficiency as if such values were known *a priori* (see Theorem 2 of [42] for details).

In this chapter, we describe two new signal-recovery methods based on the SCAD

function [104, 105]. The first method is used to solve problem (QP_λ) in (1.8) while the second one is used to solve problem (BP_δ) in (1.7). Both methods are based on the update formula in (1.22) and employ convex approximating functions of the SCAD function to render the computation of the local minimizer tractable. In the proposed sequential convex formulation (SCF) methods, we use new efficient second-order solvers (SOSs) that are applicable for the recovery of sparse signals from Gaussian ensembles. In Sec. 2.2, piecewise-linear and quadratic approximating functions of the SCAD function are presented and several results pertaining to their applicability to SCF methods are obtained. In Secs. 2.3 and 2.4, the proposed RLS and basis pursuit (BP) methods based on the SCAD function are described, respectively. In Sec. 2.5, simulation results for the proposed and corresponding competing methods are presented. In Sec. 2.6, we draw conclusions.

2.2 Convex Approximating Functions

It is assumed in this chapter that $P_\epsilon(\boldsymbol{x})$ is of the form given in (1.2), $\boldsymbol{w}$ is a column vector of n ones, $p_\epsilon(|x_i|)$ is defined by (1.6), and the gradient vector of $P_\epsilon(\boldsymbol{x})$ is given by

$$\boldsymbol{\nabla} P_\epsilon(\boldsymbol{x}) = \left[\frac{d}{dx_1}\left[p_\epsilon(|x_1|)\right] \quad \cdots \quad \frac{d}{dx_n}\left[p_\epsilon(|x_n|)\right]\right]^T \tag{2.1}$$

where

$$\frac{d}{dx_i}\left[p_\epsilon\left(|x_i|\right)\right] = \frac{d|x_i|}{dx_i} p'_\epsilon(|x_i|) \tag{2.2}$$

and

$$p'_\epsilon(|x_i|) = \begin{cases} \epsilon, & |x_i| \le \epsilon \\ \frac{(\alpha\epsilon - |x_i|)}{\alpha - 1}, & \epsilon < |x_i| \le \alpha\epsilon \\ 0, & |x_i| > \alpha\epsilon \end{cases} \tag{2.3}$$

Function $p_\epsilon\left(|x_i|\right)$ is not differentiable at $x_i = 0$ because $\frac{d|x_i|}{dx_i}$ is undefined. Without loss of generality, it is assumed that $x_i \neq 0$ in (2.2) unless otherwise specified and in such a case, (2.2) reduces to

$$\frac{d}{dx_i}\left[p_\epsilon\left(|x_i|\right)\right] = \text{sign}(x_i) p'_\epsilon(|x_i|) \tag{2.4}$$

Note that the SCAD function as defined in (1.6) is a member of class $\mathcal{N}$ because

$$\frac{d^2}{dx_i^2}\left[p_\epsilon(|x_i|)\right] = \text{sign}(x_i)p''_\epsilon(|x_i|) \tag{2.5}$$

where

$$p''_\epsilon(|x_i|) = \begin{cases} 0, & |x_i| \leq \epsilon \vee |x_i| > \alpha\epsilon \\ \frac{\alpha\epsilon - 1}{\alpha - 1}, & \epsilon < |x_i| \leq \alpha\epsilon \end{cases} \tag{2.6}$$

can assume both positive and negative values. From (1.2) and (2.5), we conclude that $P_\epsilon(\boldsymbol{x})$ is a nonconvex function because the Hessian matrix of $P_\epsilon(\boldsymbol{x})$ is diagonal with entries given by (2.5) that can assume both positive and negative values. To render minimization of $P_\epsilon(\boldsymbol{x})$ tractable, let $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ denote an approximation of $P_\epsilon(\boldsymbol{x})$ given by

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \sum_{i=1}^{n} \hat{p}_{\epsilon,x_i^{(k)}}(x_i) \tag{2.7}$$

where convex function $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ denotes an approximation of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(k)}|$. We work with SCF methods that are based on approximating functions that possess the monotonic decreasing property (MDP) stated in the following definition. This property is important because it implies that the sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.22) will converge.

Definition 2.1 (Monotonic Decreasing Property)**.** *A convex approximating function $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ is said to have the MDP at $x_i = |x_i^{(k)}|$ if and only if the conditions*

$$\hat{p}_{\epsilon,x_i^{(k)}}(x_i) \geq p_\epsilon(|x_i|), \qquad \forall x_i \in \mathbb{R} \wedge x_i \neq x_i^{(k)} \tag{2.8a}$$

and

$$\hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)}) = p_\epsilon(|x_i^{(k)}|) \tag{2.8b}$$

hold true. □

SCF methods based on functions with the MDP are applicable to the solution of nonconvex signal recovery problems. From (2.8a) and (2.8b), we have

$$\sum_{i=1}^{n} \hat{p}_{\epsilon,x_i^{(k)}}(x_i) \geq \sum_{i=1}^{n} p_\epsilon(|x_i|), \qquad \forall x_i \in \mathbb{R} \tag{2.9}$$

and

$$\sum_{i=1}^{n} \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)}) = \sum_{i=1}^{n} p_\epsilon(|x_i^{(k)}|) \tag{2.10}$$

Combining (1.2), (2.7), (2.9), and (2.10), we obtain

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \geq P_\epsilon(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathbb{R}^n \wedge \boldsymbol{x} \neq \boldsymbol{x}^{(k)} \tag{2.11}$$

and

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k)}) = P_\epsilon(\boldsymbol{x}^{(k)}) \tag{2.12}$$

Therefore, the use of convex approximating functions with the MDP implies that the conditions in (2.11) and (2.12) hold for function $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$.

If we suppose that we can find convex approximating functions that have the MDP at $|x_i^{(k)}|$, then by applying the update formula in (1.22) with $\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$, the inequalities

$$P_\epsilon(\boldsymbol{x}^{(k+1)}) \leq \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k+1)}) < \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k)}) \leq P_\epsilon(\boldsymbol{x}^{(k)}) \tag{2.13}$$

hold true and the sequence $\left\{P_\epsilon(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ is deemed a monotonically decreasing sequence. In addition, such sequence is bounded because function $P_\epsilon(\boldsymbol{x})$ is bounded from below by zero. Therefore, $\left\{P_\epsilon(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ is convergent because every bounded monotonic decreasing sequence is convergent (see monotonic sequence theorem, p. 710 of [100]).

In order to solve the convex subproblems in (1.22), we choose simple approximating functions such as the piecewise-linear and quadratic functions in the following subsections.

2.2.1 Quadratic approximation

To obtain a quadratic approximation that is applicable to SCF methods, we consider a quadratic function of the form

$$q_{x_i^{(k)}}(x_i) = a_{x_i^{(k)}} x_i^2 + b_{x_i^{(k)}} x_i + c_{x_i^{(k)}} \tag{2.14}$$

where $a_{x_i^{(k)}} \neq 0$ and $b_{x_i^{(k)}}$, $c_{x_i^{(k)}}$ are coefficients whose values are dependent on the value of $|x_i^{(k)}|$. The stationary point x_* of the quadratic function is given by

$$x_* = -\frac{b_{x_i^{(k)}}}{2a_{x_i^{(k)}}} \tag{2.15}$$

Because $p_\epsilon(|x_i|)$ in (1.6) is an even function, we conclude that the quadratic function in (2.14) must also be an even function so that the condition

$$q_{x_i^{(k)}}(-x_i) = q_{x_i^{(k)}}(x_i) = p_\epsilon(|x_i|) \tag{2.16}$$

is satisfied. Such a symmetry of $q_{x_i^{(k)}}(x_i)$ implies that $x_* = 0$. Therefore, from (2.15), we obtain

$$b_{x_i^{(k)}} = 0 \tag{2.17}$$

since $a_{x_i^{(k)}} \neq 0$. By letting $x_i = x_i^{(k)}$ in (2.16), we obtain

$$q_{x_i^{(k)}}(-x_i^{(k)}) = q_{x_i^{(k)}}(x_i^{(k)}) = p_\epsilon(|x_i^{(k)}|) \tag{2.18}$$

or equivalently

$$q_{x_i^{(k)}}(|x_i^{(k)}|) = p_\epsilon(|x_i^{(k)}|) \tag{2.19}$$

because $q_{x_i^{(k)}}(x_i)$ is an even function. Differentiating both sides of (2.19) and solving for coefficient $a_{x_i^{(k)}}$, we obtain

$$a_{x_i^{(k)}} = \frac{1}{2|x_i^{(k)}|}\, p'_\epsilon(|x_i^{(k)}|) \tag{2.20}$$

and by solving for coefficient $c_{x_i^{(k)}}$ in (2.19), we obtain

$$c_{x_i^{(k)}} = p_\epsilon(|x_i^{(k)}|) - \frac{1}{2|x_i^{(k)}|}\, p'_\epsilon(|x_i^{(k)}|)\ (|x_i^{(k)}|)^2 \tag{2.21}$$

Therefore, by combining (2.17), (2.20), and (2.21) and letting $\hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i) = q_{x_i^{(k)}}(x_i)$, the quadratic function in (2.14) can be written as

$$\hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i) = \frac{1}{2|x_i^{(k)}|}\left[x_i^2 - (|x_i^{(k)}|)^2\right] p'_\epsilon(|x_i^{(k)}|) + p_\epsilon(|x_i^{(k)}|) \tag{2.22}$$

Function $\hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i)$ defines a quadratic approximation (QA) of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(k)}|$. This approximating function is convex because coefficient $a_{x_i^{(k)}}$ in (2.20) can only assume positive values (see (2.3)) and the approximation is also a smooth function of x_i. The QA of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(0)}|$ for $|x_i^{(0)}| = 5/2$ and $|x_i^{(0)}| = 3/2$ is plotted in Fig. 2.1.

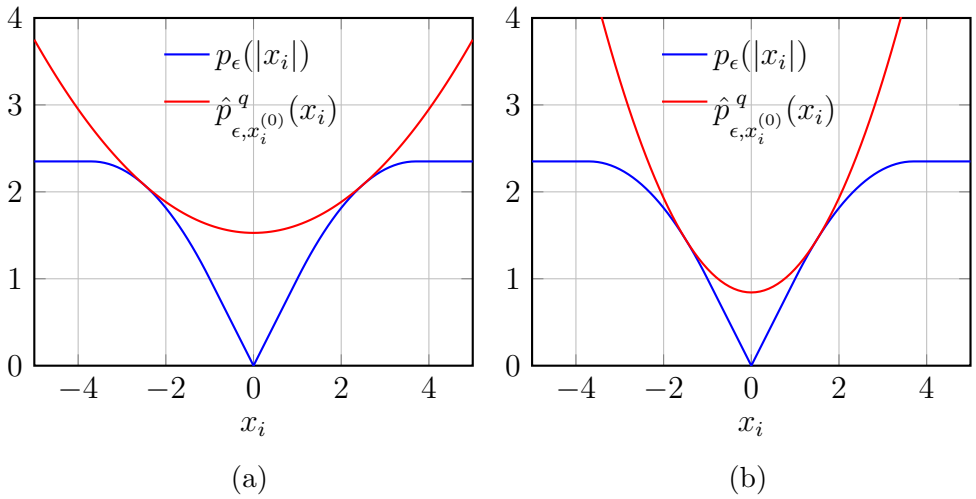


Figure 2.1: QA of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(0)}|$: (a) $|x_i^{(0)}| = 5/2$ and (b) $|x_i^{(0)}| = 3/2$.

We now show that the QA is applicable to the solution of nonconvex recovery problems because it can be used in conjunction with SCF methods.

Proposition 2.1 (Monotonic Decreasing Quadratic Approximation)**.** *The convex approximating function* $\hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i)$ *in* (2.22) *has the MDP at* $x_i = |x_i^{(k)}|$ *for any value of* $|x_i^{(k)}|$ *except* 0.

Proof. Let $x_i^{(k)} \neq 0$. From (2.18), we obtain

$$\hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i^{(k)}) = p_\epsilon(|x_i^{(k)}|) \tag{2.23}$$

and hence the condition in (2.8b) is satisfied for function $\hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i)$. Now let

$$g(x_i) = \hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i) - p_\epsilon(|x_i|) \tag{2.24}$$

Functions $\hat{p}^{\,q}_{\epsilon,x_i^{(k)}}(x_i)$ and $p_\epsilon(|x_i|)$ are even as is their summation. Hence, $g(x_i)$ is an

even function. The first-order derivative of $g(x_i)$ can be written as

$$\frac{d}{dx_i}\left[g(x_i)\right] = x_i \left\{ \frac{1}{|x_i^{(k)}|}\, p'_\epsilon(|x_i^{(k)}|) - \frac{d}{dx_i}\left[p_\epsilon(|x_i|)\right] \frac{1}{x_i} \right\} \tag{2.25}$$

Without loss of generality, consider the case where x_i can only assume positive values in (2.25), i.e., let $x_i \in (0, \infty)$. From (1.6), we note that $p_\epsilon(|x_i|)$ is a nondecreasing function of x_i. Hence, $\frac{d}{dx_i}\left[p_\epsilon(|x_i|)\right] \frac{1}{x_i}$ is a nonincreasing function of x_i. By using this property in (2.25), we obtain

$$\frac{d}{dx_i} g(x_i) \leq 0 \qquad \text{for } x_i \in (0,\ |x_i^{(k)}|) \tag{2.26a}$$

and

$$\frac{d}{dx_i} g(x_i) \geq 0 \qquad \text{for } x_i \in (|x_i^{(k)}|,\ \infty) \tag{2.26b}$$

From 2.26a and (2.26b), we find that function $g(x_i)$ must be (1) nonincreasing for $x_i \in (0,\ |x_i^{(k)}|)$, and (2) nondecreasing for $x_i \in (|x_i^{(k)}|,\ \infty)$. Therefore, $g(x_i)$ has a minimum at $|x_i^{(k)}|$. By combining (2.23) and (2.24), we note that function $g(x_i)$ assumes the value of zero at $|x_i^{(k)}|$. Since $g(x_i)$ is an even function, the above analysis holds true in the case where $x_i \in (-\infty, 0)$. Therefore,

$$g(x_i) \geq 0 \qquad \forall x_i \in (-\infty, 0) \cup (0, \infty) \tag{2.27}$$

and the condition in (2.8a) is satisfied for function $\hat{p}^{\,q}_{\epsilon, x_i^{(k)}}(x_i)$ when $x_i^{(k)} \neq 0$.

In summary, the conditions in 2.8a and (2.8b) are satisfied for function $\hat{p}^{\,q}_{\epsilon, x_i^{(k)}}(x_i)$. Therefore, $\hat{p}^{\,q}_{\epsilon, x_i^{(k)}}(x_i)$ has the MDP at $x_i = |x_i^{(k)}|$ for any value of $|x_i^{(k)}|$ except 0. □

The computation of the QA in (2.22) is problematic when $|x_i^{(k)}|$ assumes a value of zero because the approximation is a function of the reciprocal of $|x_i^{(k)}|$ which is undefined at $|x_i^{(k)}| = 0$. We address this issue in Sec. 2.3 when describing the proposed SCF method based on such an approximation.

2.2.2 Piecewise-linear approximation

The first-order Taylor series approximations of $p_\epsilon(x_i)$ at $x_i = |x_i^{(k)}|$ and $x_i = -|x_i^{(k)}|$, namely,

$$p_\epsilon(x_i) \approx p_\epsilon(|x_i^{(k)}|) + \left(x_i - |x_i^{(k)}|\right) p'_\epsilon(|x_i^{(k)}|) \tag{2.28}$$

and

$$p_\epsilon(-x_i) \approx p_\epsilon(|x_i^{(k)}|) + \left(-x_i - |x_i^{(k)}|\right) p'_\epsilon(|x_i^{(k)}|) \tag{2.29}$$

respectively, define convex approximations. By combining (2.28) and (2.29), we obtain

$$\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) = p_\epsilon(|x_i^{(k)}|) + \left(|x_i| - |x_i^{(k)}|\right) p'_\epsilon(|x_i^{(k)}|) \tag{2.30}$$

which defines a piecewise-linear approximation (PLA) of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(k)}|$. This approximating function is convex because the functions in (2.28) and (2.29) are convex. In addition, the approximating function is a nonsmooth function of x_i. The PLA of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(0)}|$ for $|x_i^{(0)}| = 5/2$ and $|x_i^{(0)}| = 3/2$ is plotted in Fig. 2.2.

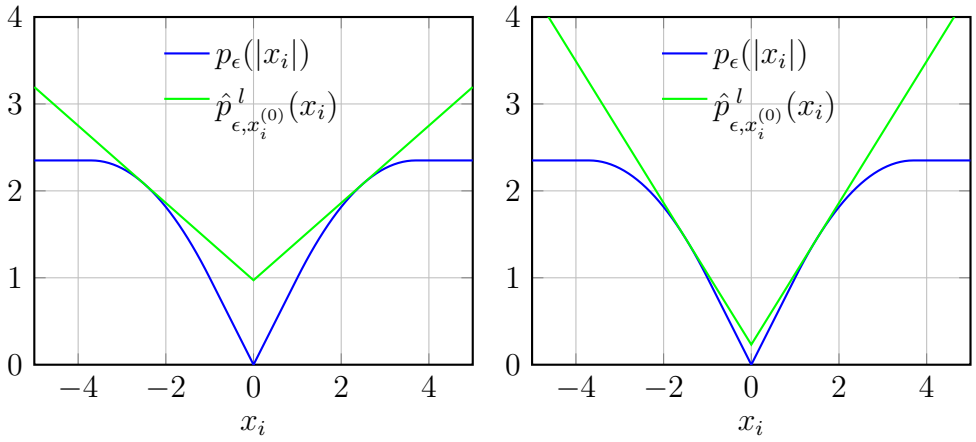


Figure 2.2: PLA of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(0)}|$: (a) $|x_i^{(0)}| = 5/2$ and (b) $|x_i^{(0)}| = 3/2$.

We now show that, like the QA, the PLA is applicable to the solution of nonconvex recovery problems because it can be used in conjunction with SCF methods.

Proposition 2.2 (Monotonic Decreasing Piecewise-Linear Approximation)**.** *The convex approximating function* $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$ *in* (2.30) *has the MDP at* $x_i = |x_i^{(k)}|$.

Proof. By letting $x_i = x_i^{(k)}$ in (2.30), we obtain

$$\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i^{(k)}) = p_\epsilon(|x_i^{(k)}|) \tag{2.31}$$

and the condition in (2.8b) is satisfied for function $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$. Without loss of generality, consider the case where $x_i \in (0, \infty)$. From (2.5), function $p_\epsilon(|x_i|)$ is concave

in such a case. A first-order necessary and sufficient condition for the concavity of $p_\epsilon(|x_i|)$ is given by

$$p_\epsilon(|x_i|) \leq p_\epsilon(|x_i^{(k)}|) + \left(|x_i| - |x_i^{(k)}|\right) p'_\epsilon(|x_i^{(k)}|) \tag{2.32}$$

(see p. 70 of [16]). Now let

$$g(x_i) = \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) - p_\epsilon(|x_i|) \tag{2.33}$$

By using (2.33) and (2.32), it can readily be shown that $g(x_i) \geq 0$. Functions $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$ and $p_\epsilon(|x_i|)$ are even and so is their summation. Hence, $g(x_i)$ is an even function and the above analysis also holds true when $x_i \in (-\infty, 0)$. Therefore,

$$g(x_i) \geq 0, \qquad \forall x_i \in (-\infty, 0) \cup (0, \infty) \tag{2.34}$$

and the condition in (2.8a) is satisfied for function $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$.

In summary, the conditions in 2.8a and (2.8b) are satisfied for function $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$. Therefore, $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$ has the MDP at $x_i = |x_i^{(k)}|$. □

The approximating function $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$ in (2.30) is in fact the best convex approximation of function $p_\epsilon(|x_i|)$ because it provides the least upper bound on $p_\epsilon(|x_i|)$. This is demonstrated in terms of the following proposition.

Proposition 2.3 (Best Convex Approximation)**.** *Let $\mathcal{A}$ denote the class of all convex approximating functions with the MDP at $x_i = |x_i^{(k)}|$ and let $\hat{p}_{\epsilon,x_i^{(k)}}(x_i) \in \mathcal{A}$ denote a convex approximating function in class $\mathcal{A}$. The condition*

$$\hat{p}_{\epsilon,x_i^{(k)}}(x_i) \geq \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) \geq p_\epsilon(|x_i|), \qquad \forall x_i \in \mathbb{R} \wedge x_i \neq x_i^{(k)} \tag{2.35}$$

holds true for the PLA of function $p_\epsilon(|x_i|)$.

Proof. Since $\hat{p}_{\epsilon,x_i^{(k)}}(x_i) \in \mathcal{A}$ and $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) \in \mathcal{A}$, it suffices to show that

$$\hat{p}_{\epsilon,x_i^{(k)}}(x_i) - \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) \geq 0, \qquad \forall x_i \in \mathbb{R} \wedge x_i \neq x_i^{(k)} \tag{2.36}$$

because in such case the condition in (2.35) is satisfied. By using (2.30) and the

property $\hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)}) = p_\epsilon(|x_i^{(k)}|)$, (2.36) is equivalent to

$$\frac{\hat{p}_{\epsilon,x_i^{(k)}}(x_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{|x_i| - |x_i^{(k)}|} \geq p'_\epsilon(|x_i^{(k)}|), \qquad \forall x_i \in \mathbb{R} \wedge x_i \neq x_i^{(k)} \tag{2.37}$$

Without loss of generality, let $x_i \in (0, \infty)$ and consider the case where $|x_i^{(k)}| < x'_i < x_i$. Because $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ is a convex function the inequality

$$\frac{\hat{p}_{\epsilon,x_i^{(k)}}(x_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{|x_i| - |x_i^{(k)}|} \geq \frac{\hat{p}_{\epsilon,x_i^{(k)}}(x'_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{x'_i - |x_i^{(k)}|} \tag{2.38}$$

holds true for $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ (see Exercise 3.1 of [16]). Because function $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ has the MDP at $x_i = |x_i^{(k)}|$ the inequality

$$\frac{\hat{p}_{\epsilon,x_i^{(k)}}(x'_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{x'_i - |x_i^{(k)}|} \geq \frac{p_\epsilon(|x'_i|) - p_\epsilon(|x_i^{(k)}|)}{x'_i - |x_i^{(k)}|} \tag{2.39}$$

holds true for $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$. Combining (2.38) and (2.39), we obtain

$$\frac{\hat{p}_{\epsilon,x_i^{(k)}}(x_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{|x_i| - |x_i^{(k)}|} \geq \frac{p_\epsilon(|x'_i|) - p_\epsilon(|x_i^{(k)}|)}{x'_i - |x_i^{(k)}|} \tag{2.40}$$

Taking the limit of the above inequality, we have

$$\begin{aligned}
\frac{\hat{p}_{\epsilon,x_i^{(k)}}(x_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{|x_i| - |x_i^{(k)}|} &\geq \lim_{x'_i \to |x_i^{(k)}|} \left[\frac{p_\epsilon(|x'_i|) - p_\epsilon(|x_i^{(k)}|)}{x'_i - |x_i^{(k)}|} \right] \\
\frac{\hat{p}_{\epsilon,x_i^{(k)}}(x_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{|x_i| - |x_i^{(k)}|} &\geq \lim_{x'_i \to |x_i^{(k)}|} \frac{\frac{d}{dx'_i}\left[p_\epsilon(|x'_i|) - p_\epsilon(|x_i^{(k)}|) \right]}{\frac{d}{dx'_i}\left[x'_i - |x_i^{(k)}| \right]} \\
\frac{\hat{p}_{\epsilon,x_i^{(k)}}(x_i) - \hat{p}_{\epsilon,x_i^{(k)}}(x_i^{(k)})}{|x_i| - |x_i^{(k)}|} &\geq p'_\epsilon(|x_i^{(k)}|)
\end{aligned} \tag{2.41}$$

A similar approach can be used to show that (2.41) holds true in the case where $x_i < x'_i < |x_i^{(k)}|$. Since $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ is even, (2.41) is applicable when $x_i \in (-\infty, 0)$. Thus, the condition in (2.37) holds true. □

A comparison between the QA and the PLA of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(0)}|$ for $|x_i^{(0)}| = 5/2$

and $|x_i^{(0)}| = 3/2$ is illustrated in Fig. 2.3. As can be seen, the PLA yields the tightest upper bound on function $p_\epsilon(|x_i|)$.

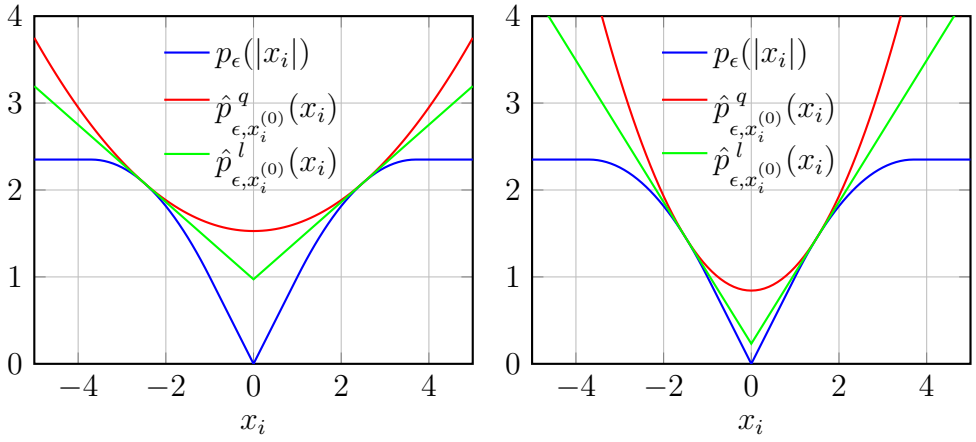


Figure 2.3: Comparison of the QA and PLA of $p_\epsilon(|x_i|)$ at $x_i = |x_i^{(0)}|$: (a) $|x_i^{(0)}| = 5/2$ and (b) $|x_i^{(0)}| = 3/2$

2.3 QA Based RLS Method

An SCF method for the solution of problem (QP_λ) in (1.8) is now described [104]. The problem can be expressed as

$$(\mathrm{QP}_\epsilon) \qquad \underset{\boldsymbol{x}}{\text{minimize}} \quad F(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + P_\epsilon(\boldsymbol{x}) \tag{2.42}$$

where parameter ϵ corresponds to the regularization term of problem (QP_ϵ) when $p_\epsilon(|x_i|)$ is given by (1.6). Parameter ϵ controls the trade-off between measurement consistency and sparsity of the solution just like parameter λ of problem (QP_λ) in (1.8) (see Sec. 2 of [42] for details). We have seen in Sec. 1.2.3 that SCF methods are applicable to the minimization of a nonconvex function $F(\boldsymbol{x})$ over a convex set $\mathcal{X}$. Thus, problem (QP_ϵ) corresponds to the problem of minimizing the nonconvex function $F(\boldsymbol{x})$ over set $\mathcal{X} = \mathbb{R}^n$. A solution can be found by letting $\boldsymbol{x}^0 \in \mathbb{R}^n$ and by applying the update formula in (1.22) with

$$\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{2.43}$$

where $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is based on the QA in (2.22). Hence, (1.22) can be rewritten as

$$\boldsymbol{x}^{(k+1)} = \arg \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\ \widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{2.44}$$

Because function $\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is based on the QA, we must have

$$\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \geq F(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathbb{R}^n \wedge \boldsymbol{x} \neq \boldsymbol{x}^{(k)}$$

and

$$\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k)}) = F(\boldsymbol{x}^{(k)})$$

Therefore,

$$F(\boldsymbol{x}^{(k+1)}) \leq \widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k+1)}) < \widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k)}) \leq F(\boldsymbol{x}^{(k)}) \tag{2.45}$$

holds and the update formula in (2.44) assures the monotonic decrease of $F(\boldsymbol{x})$. A suitable stopping criterion for the computation of $\boldsymbol{x}^{(k+1)}$ is

$$\left\| F(\boldsymbol{x}^{(k+1)}) - F(\boldsymbol{x}^{(k)}) \right\|_2 \leq \varepsilon_c \tag{2.46}$$

where ε_c is a small positive constant since $\left\{F(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ is a convergent sequence (see Sec. 2.2).

2.3.1 Dimensionality Reduction

Function $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is based on the QA and computation of the update formula in (2.44) is problematic when a coordinate of $\boldsymbol{x}^{(k)}$ assumes the value of zero. In such a case, $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is undefined (see Sec. 2.2.1). Let

$$\mathcal{I}_m = \{1,\ 2,\ \ldots,\ m\} \qquad \text{and} \qquad \mathcal{I}^{(k)} = \left\{i : |x_i^{(k)}| \geq \varepsilon_r\right\}$$

denote sets of integers of cardinality m and n_k, respectively, where ε_r is a positive constant. Also let $\boldsymbol{A}(\mathcal{I}_m, \mathcal{I}^{(k)})$ denote the submatrix that consists of the rows of $\boldsymbol{A}$ indexed by $\mathcal{I}_m$ and the columns indexed by $\mathcal{I}^{(k)}$. Similarly, let $\boldsymbol{x}^{(k)}(\mathcal{I}^{(k)})$ denotes the subvector that consists of the columns of $\boldsymbol{x}^{(k)}$ indexed by $\mathcal{I}^{(k)}$. In such a case, function $\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ in (2.43) is redefined as

$$\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) = \frac{1}{2}||\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \boldsymbol{b}||_2^2 + \widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) \tag{2.47}$$

where $\widetilde{\boldsymbol{A}} = \boldsymbol{A}(\mathcal{I}_m, \mathcal{I}^{(k)})$, $\tilde{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(k)}(\mathcal{I}^{(k)})$, $\tilde{\boldsymbol{x}} \in \mathbb{R}^{n_k}$, and

$$\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) = \sum_{i \in \mathcal{I}^{(k)}} \hat{p}^{\,q}_{\epsilon,\tilde{x}_i^{(k)}}(\tilde{x}_i) \tag{2.48}$$

The new update formula applied to function $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ in (2.47) is given by

$$\tilde{\boldsymbol{x}}^{(k+1)} = \arg \underset{\tilde{\boldsymbol{x}}}{\text{minimize}}\ \widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) \tag{2.49}$$

and the issue of computing the update formula in (2.44) when $x_i^{(k)} = 0$ can be overcome by performing Algorithm 2.1.

Input: $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, $\varepsilon_r, \varepsilon_c > 0$
Output: $\boldsymbol{x}^{(k+1)}$
$k = 0$;
$\mathcal{I}^{(k)} = \{1, \ \ldots, \ n\}$;
$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}$;
repeat
  $k = k + 1$;
  Compute: $\mathcal{I}^{(k)}$, $\boldsymbol{A}(\mathcal{I}_m, \mathcal{I}^{(k)})$, $\boldsymbol{x}^{(k)}(\mathcal{I}^{(k)})$;
  $x_i^{(k)} = 0$ for $i \notin \mathcal{I}^{(k)}$;
  Apply the update formula in (2.49) to obtain $\tilde{\boldsymbol{x}}^{(k+1)}$;
  $x_i^{(k+1)} = \tilde{x}_i^{(k+1)}$ for $i \in \mathcal{I}^{(k)}$;
until $\left\| F(\boldsymbol{x}^{(k+1)}) - F(\boldsymbol{x}^{(k)}) \right\|_2 \leq \varepsilon_c$;

Algorithm 2.1: Dimensionality Reduction

Because each coordinate of the current point $\tilde{\boldsymbol{x}}^{(k)}$ can never assume the value of zero, function $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ is always well-defined and the update formula in (2.49) can be used for the computation of the next point $\tilde{\boldsymbol{x}}^{(k+1)}$. In addition, on the basis of (2.45), we conclude that the minimization of $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ has a sparse solution and the number of nonzero coordinates of $\boldsymbol{x}^{(k+1)}$ is, therefore, less or equal to the number of nonzero coordinates of $\boldsymbol{x}^{(k)}$. As a result, the dimension of the convex subproblem in (2.49) can be reduced from step k to step $k+1$ because $|\mathcal{I}^{(0)}| \geq |\mathcal{I}^{(1)}| \geq \ \cdots \ \geq |\mathcal{I}^{(k)}|$ where $|\mathcal{I}^{(k)}|$ denotes the cardinality of set $\mathcal{I}^{(k)}$ and, therefore, $n_1 \geq n_2 \geq \ldots \geq n_k$. If parameter ε_r in Algorithm 2.1 is set too large, a coordinate of $\boldsymbol{x}^{(k)}$ can be incorrectly updated to zero. This issue can be circumvented by selecting a suitable value of ε_r.

2.3.2 Proposed SOS

Function $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ in (2.48) can be written as

$$\begin{aligned}\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) &= \sum_{i\in\mathcal{I}^{(k)}} \hat{p}^{\,q}_{\epsilon,\tilde{x}_i^{(k)}}(\tilde{x}_i) \\ &= \frac{1}{2}\mathbf{1}^T\left(\widetilde{\boldsymbol{X}}^2 - \left|\widetilde{\boldsymbol{X}}^{(k)}\right|^2\right)\boldsymbol{p}\end{aligned} \tag{2.50}$$

where $\mathbf{1}$ is a column vector of n_k ones, matrices $\widetilde{\boldsymbol{X}}$ and $\left|\widetilde{\boldsymbol{X}}^{(k)}\right|$ are given by

$$\widetilde{\boldsymbol{X}} = \operatorname{diag}\left\{\tilde{x}_1,\ \tilde{x}_2,\ \ldots,\ \tilde{x}_{n_k}\right\} \qquad \text{and} \qquad \left|\widetilde{\boldsymbol{X}}^{(k)}\right| = \operatorname{diag}\left\{|\tilde{x}_i^{(k)}|\right\}_{i\in\mathcal{I}^{(k)}}$$

and $\boldsymbol{p}$ is a column vector of length q given by

$$\boldsymbol{p} = \left[\left|\widetilde{\boldsymbol{X}}^{(k)}\right|^{-1}\boldsymbol{\nabla}P_\epsilon(\tilde{\boldsymbol{x}}^{(k)})\right] \tag{2.51}$$

Hence, function $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ in (2.47) can be written as

$$\begin{aligned}\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) &= \frac{1}{2}||\widetilde{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{b}||_2^2 + \widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) \\ &= \frac{1}{2}\tilde{\boldsymbol{x}}^T\widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}^T\widetilde{\boldsymbol{A}}^T\boldsymbol{b} + \frac{1}{2}\mathbf{1}^T\left(\widetilde{\boldsymbol{X}}^2 - \left|\widetilde{\boldsymbol{X}}^{(k)}\right|^2\right)\boldsymbol{p}\end{aligned} \tag{2.52}$$

Terms that do not involve $\tilde{\boldsymbol{x}}$ are constant in the minimization of $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ and can be dropped as they do not change the solution. Therefore, each convex subproblem in (2.49) is equivalent to

$$\underset{\tilde{\boldsymbol{x}}}{\text{minimize}}\ \widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) \tag{2.53}$$

where

$$\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) = \frac{1}{2}\tilde{\boldsymbol{x}}^T\widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}^T\widetilde{\boldsymbol{A}}^T\boldsymbol{b} + \frac{1}{2}\mathbf{1}^T\widetilde{\boldsymbol{X}}^2\boldsymbol{p}$$

The gradient vector and Hessian matrix of the problem in (2.53) are given by

$$\boldsymbol{\nabla}\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) = \widetilde{\boldsymbol{A}}^T(\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \boldsymbol{b}) + \widetilde{\boldsymbol{X}}\boldsymbol{p} \qquad \text{and} \qquad \boldsymbol{\nabla}^2\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) = \widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}} + \operatorname{diag}(\boldsymbol{p})$$

respectively. The update formula for the minimization of $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ can be obtained as

$$\tilde{\boldsymbol{x}}_{i+1} = \tilde{\boldsymbol{x}}_i - \boldsymbol{\nabla}^2\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}_i)\boldsymbol{\nabla}\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}_i) \tag{2.54}$$

by applying Newton's method (see Sec. 5.3 of [4]). Because $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ is a quadratic convex function, sequence $\{\tilde{\boldsymbol{x}}_i\}_{i\in\mathbb{N}}$ in (2.54) converges to the minimizer $\tilde{\boldsymbol{x}}^*$ of $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$ in just one step from any initial point $\tilde{\boldsymbol{x}}_0$. So we let $\tilde{\boldsymbol{x}}_0 = \mathbf{0}$ in (2.54) where $\mathbf{0}$ is a column vector of n_k zeros and the next point $\tilde{\boldsymbol{x}}_1$ is the minimizer of $\widehat{F}_{\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})$. Therefore, we obtain the analytical solution

$$\tilde{\boldsymbol{x}}^* = \boldsymbol{M}^{-1}\left[\widetilde{\boldsymbol{A}}^T(\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \boldsymbol{b}) + \widetilde{\boldsymbol{X}}\boldsymbol{p}\right] \tag{2.55}$$

where

$$\boldsymbol{M} = \operatorname{diag}(\boldsymbol{p}) + \widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}} \tag{2.56}$$

is an $n_k \times n_k$ matrix.

The computation of (2.55) is problematic because it requires the inversion of a large $n_k \times n_k$ matrix. Fortunately, matrix $\boldsymbol{M}$ has a special structure that facilitates the efficient computation of $\boldsymbol{M}^{-1}$. Suppose for a moment that the magnitudes of the coordinates of vector $\boldsymbol{p}$ in (2.51) are always greater than zero. In such a case, we note that (1) matrix $\operatorname{diag}(\boldsymbol{p})$ in (2.56) is invertible and the operation $[\operatorname{diag}(\boldsymbol{p})]^{-1}$ can be computed analytically by taking the reciprocal of each coordinate of vector $\boldsymbol{p}$, and (2) The number of rows of matrix $\widetilde{\boldsymbol{A}}$ in (2.56) is much smaller than its number of columns, e.g., when k is not large in (2.49), m is much smaller than n_k. In such a case, we can employ the Sherman-Morrison-Woodbury formula [54]

$$\boldsymbol{M}^{-1} = \operatorname{diag}(\boldsymbol{p})^{-1} - \operatorname{diag}(\boldsymbol{p})^{-1}\boldsymbol{A}^T\left(\boldsymbol{I} + \boldsymbol{A}\operatorname{diag}(\boldsymbol{p})^{-1}\boldsymbol{A}^T\right)^{-1}\boldsymbol{A}\operatorname{diag}(\boldsymbol{p})^{-1} \tag{2.57}$$

because $\operatorname{diag}(\boldsymbol{p})^{-1}$ can be easily computed and because the effort involved in the computation of the $m \times m$ matrix inverse

$$\left(\boldsymbol{I} + \boldsymbol{A}\operatorname{diag}(\boldsymbol{p})^{-1}\boldsymbol{A}^T\right)^{-1}$$

is small relative to the effort involved in the computation of $\boldsymbol{M}^{-1}$ since $m \ll n_k$. In the case where a coordinate of vector $\boldsymbol{p}$ is zero, we add a small positive constant ε_d to such coordinate, e.g., of the order of 10^{-9}, so that matrix $\operatorname{diag}(\boldsymbol{p})$ is always invertible and (2.57) can be employed for the computation of $\boldsymbol{M}^{-1}$.

2.3.3 Continuation procedure

It has been observed that the convergence rate of sequence $\left\{F(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ in (2.46) slows for decreasing values of ϵ [104]. Reduced computational cost can be achieved by using a continuation procedure where several (QP_ϵ) problems are solved in sequence for decreasing values of ϵ in the same way as is done in recent RLS methods [8,43,68]. In the proposed continuation procedure, the solution obtained for the previous (QP_ϵ) problem is used as the initial point for the solution of the next (QP_ϵ) problem. Thus, the convergence rate of sequence $\left\{F(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ for each (QP_ϵ) problem is improved due to an appropriate initialization.

In summary, the proposed SCF method is efficient because (1) the convex subproblems can be solved by an efficient SOS, (2) the computation of $\boldsymbol{x}^{(k)}$ in Algorithm 2.1 entails reduced subproblem sizes for increasing values of k, and (3) the convergence rate of sequence $\left\{F(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ can be improved by using a continuation procedure.

2.4 PLA Based BP Method

An SCF method for the solution of problem (BP_δ) in (1.7) is now described [105]. Using the notation of Sec. 1.2.3, problem (BP_δ) entails minimizing nonconvex function $F(\boldsymbol{x}) = P_\epsilon(\boldsymbol{x})$ over convex set $\mathcal{X} = \mathcal{K}$, where $\mathcal{K}$ is the closed ball in $\mathbb{R}^n$ under an affine mapping given by

$$\mathcal{K} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \delta\} \tag{2.58}$$

We can find a solution by letting $\boldsymbol{x}^0 \in \mathcal{K}$ and then applying the update formula in (1.22) with $\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$, where $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is based on the PLA in (2.30). Hence, (1.22) can be written as

$$\boldsymbol{x}^{(k+1)} = \arg \underset{\boldsymbol{x}\in\mathcal{K}}{\text{minimize}}\ \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{2.59}$$

where function $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is given by

$$\begin{aligned}\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) &= \sum_{i=1}^{n} \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) \\ &= \sum_{i=1}^{n} \left[p_\epsilon(|x_i^{(k)}|) + \left(|x_i| - |x_i^{(k)}|\right) p'_\epsilon(|x_i^{(k)}|) \right]\end{aligned}$$

The use of the update formula in (2.59) assures the monotonic decrease of function $P_\epsilon(\boldsymbol{x})$ of problem (BP_δ) because the use of the PLA implies that the inequalities in (2.13) hold. A suitable stopping criterion for the computation of $\boldsymbol{x}^{(k+1)}$ is

$$\left\| P_\epsilon(\boldsymbol{x}^{(k+1)}) - P_\epsilon(\boldsymbol{x}^{(k)}) \right\|_2 \leq \varepsilon_c \tag{2.60}$$

where ε_c is a small positive constant since $\left\{ P_\epsilon(\boldsymbol{x}^{(k)}) \right\}_{k\in\mathbb{N}}$ is a convergent sequence (see Sec. 2.2).

2.4.1 Proposed SOS

In the minimization of $\widehat{P}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})$, terms that do not involve $\boldsymbol{x}$ are constant and can be dropped as they do not change the solution. Therefore, each convex subproblem in (2.59) can be written as

$$\underset{\boldsymbol{x}\in\mathcal{K}}{\text{minimize}}\ \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{2.61}$$

where

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \sum_{i=1}^{n} p'_\epsilon(|x_i^{(k)}|)|x_i|$$

Since function $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is based on the PLA, the computation of the update formula in (2.59) is problematic because $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is a nonsmooth function with respect to $\boldsymbol{x}$ and the convex subproblems are nonsmooth as a result. We address this issue by (1) introducing an upper bound u_i for each term of the summation in (2.61), (2) splitting each absolute value $|x_i|$ in terms of this upper bound as $|x_i| = (\pm x_i - u_i)$, and (3) creating inequality constraints in terms of each positive and negative parts of the upper bounded terms of the summation. Therefore, by letting $\mathcal{I}_n = \{1,\ \ldots,\ n\}$ and $\mathcal{J}_n = \{n+1,\ \ldots,\ 2n\}$ denote sets of n integers, we conclude that the problem

in (2.61) is equivalent to the *smooth* convex optimization problem

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathcal{K},\boldsymbol{u}}{\text{minimize}} \quad & \sum_{i=1}^{n} u_i \\ \text{subject to:} \quad & p'_\epsilon(|x_i^{(k)}|)(x_i - u_i) \leq 0, \qquad i \in \mathcal{I}_n \\ & -p'_\epsilon(|x_i^{(k)}|)(-x_i - u_i) \geq 0, \qquad i \in \mathcal{I}_n \end{aligned} \tag{2.62}$$

By using the definition of $p'_\epsilon(|x_i|)$ in (2.3), we can combine the inequality constraints of the problem in (2.62) to obtain the equivalent problem

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathcal{K},\boldsymbol{u}}{\text{minimize}} \quad & \sum_{i=1}^{n} u_i \\ \text{subject to:} \quad & l_j(|x_i^{(k)}|, x_i, u_i) \leq 0, \quad (i \in \mathcal{I}_n) \wedge [j \in (\mathcal{I}_n \cup \mathcal{J}_n)] \end{aligned} \tag{2.63}$$

where $l_j(|x_i^{(k)}|, x_i, u_i)$ for $j \in (\mathcal{I}_n \cup \mathcal{J}_n)$ are $2n$ linear constraints of the form given by

$$l_j(|x_i^{(k)}|, x_i, u_i) = \begin{cases} \epsilon x_i - u_i, & |x_i^{(k)}| \leq \epsilon \\ \frac{\left(\alpha\epsilon - |x_i^{(k)}|\right)}{\alpha - 1} x_i - u_i, & \epsilon < |x_i^{(k)}| \leq \alpha\epsilon \\ -u_i, & |x_i^{(k)}| > \alpha\epsilon \end{cases} \tag{2.64a}$$

for $(i = j) \wedge (j \in \mathcal{I}_n)$ and

$$l_j(|x_i^{(k)}|, x_i, u_i) = \begin{cases} -\epsilon x_i - u_1, & |x_i^{(k)}| \leq \epsilon \\ -\frac{\left(\alpha\epsilon - |x_i^{(k)}|\right)}{\alpha - 1} x_i - u_i, & \epsilon < |x_i^{(k)}| \leq \alpha\epsilon \\ 0, & |x_i^{(k)}| > \alpha\epsilon \end{cases} \tag{2.64b}$$

for $(i = j - n) \wedge (j \in \mathcal{J}_n)$.

Consider the linear constraints in (2.64) and let $\tilde{\boldsymbol{x}} = \begin{bmatrix} x_1 & \cdots & x_n & u_1 & \cdots & u_n \end{bmatrix}^T$ denote the new optimization variable of the problem in (2.63). Whenever $|x_i^{(k)}| \leq \epsilon$, we write the constraints $l_j(x_i, u_i)$ for two indices j in terms of the new variable $\tilde{\boldsymbol{x}}$ and $2n$-length column vectors $\boldsymbol{q}_j$ for $j \in \mathcal{I}_n$ and $\bar{\boldsymbol{q}}_j$ for $j \in \mathcal{J}_n$ with entries $q_{j,l}$ and $\bar{q}_{j,l}$ for

$l \in (\mathcal{I}_n \cup \mathcal{J}_n)$ given by

$$q_{j,l} = \begin{cases} \epsilon, & l = j \\ -1, & l = j+n \\ 0, & \text{otherwise} \end{cases} \qquad \text{and} \qquad \bar{q}_{j,l} = \begin{cases} -\epsilon, & l = j-n \\ -1, & l = j \\ 0, & \text{otherwise} \end{cases}$$

Similarly, whenever $\epsilon < |x_i^{(k)}| \leq \alpha\epsilon$, we can write $l_j(x_i, u_i)$ in terms of $\tilde{\boldsymbol{x}}$ and $2n$-length column vectors $\boldsymbol{r}_j$ for $j \in \mathcal{I}_n$ and $\bar{\boldsymbol{r}}_j$ for $j \in \mathcal{J}_n$ with entries $r_{j,l}$ and $\bar{r}_{j,l}$ for $l \in (\mathcal{I}_n \cup \mathcal{J}_n)$ given by

$$r_{j,l} = \begin{cases} \frac{\alpha\epsilon - |x_i^{(k)}|}{\alpha - 1}, & l = j \\ -1, & l = j+n \\ 0, & \text{otherwise} \end{cases} \qquad \text{and} \qquad \bar{r}_{j,l} = \begin{cases} -\frac{\alpha\epsilon - |x_i^{(k)}|}{\alpha - 1}, & l = j-n \\ -1, & l = j \\ 0, & \text{otherwise} \end{cases}$$

For the last possible case, whenever $|x_i^{(k)}| > \alpha\epsilon$, we can write $l_j(x_i, u_i)$ in terms of $\tilde{\boldsymbol{x}}$ and a $2n$-length column vector $\boldsymbol{s}_j$ for $j \in \mathcal{I}_n$ with entries $s_{j,l}$ given by

$$s_{j,l} = \begin{cases} -1, & l = j+n \\ 0, & \text{otherwise} \end{cases}$$

By letting $\boldsymbol{C}^{(k)} = \begin{bmatrix} \boldsymbol{c}_1^T & \cdots & \boldsymbol{c}_n^T & \boldsymbol{c}_{n+1}^T & \cdots & \boldsymbol{c}_{2n}^T \end{bmatrix}^T$ denote a $2n \times 2n$ matrix where the $2n$-length column vectors $\boldsymbol{c}_i$ and $\boldsymbol{c}_{i+n}$ for $i \in \mathcal{I}_n$ are given by

$$\boldsymbol{c}_i = \begin{cases} \boldsymbol{q}_i, & |x_i^{(k)}| \leq \epsilon \\ \boldsymbol{r}_i, & \epsilon < |x_i^{(k)}| \leq \alpha\epsilon \\ \boldsymbol{s}_i, & |x_i^{(k)}| > \alpha\epsilon \end{cases} \qquad \text{and} \qquad \boldsymbol{c}_{i+n} = \begin{cases} \bar{\boldsymbol{q}}_i, & |x_i^{(k)}| \leq \epsilon \\ \bar{\boldsymbol{r}}_i, & \epsilon < |x_i^{(k)}| \leq \alpha\epsilon \\ \boldsymbol{0}_{2n\times 1}, & |x_i^{(k)}| > \alpha\epsilon \end{cases} \tag{2.65}$$

and by letting $\widetilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{0}_{q\times n} \end{bmatrix}$ and $\boldsymbol{e} = \begin{bmatrix} \boldsymbol{0}_{n\times 1} & \boldsymbol{1}_{n\times 1} \end{bmatrix}^T$ in the problem in (2.63), the update formula in (2.59) can be written in the equivalent form

$$\tilde{\boldsymbol{x}}^{(k+1)} = \arg \underset{\tilde{\boldsymbol{x}} \in (\widetilde{\mathcal{K}} \cap \widetilde{\mathcal{M}})}{\text{minimize}} \; \boldsymbol{e}^T \tilde{\boldsymbol{x}} \tag{2.66}$$

where

$$\widetilde{\mathcal{K}} = \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n} : ||\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \boldsymbol{b}||_2 \leq \delta\right\} \qquad \text{and} \qquad \widetilde{\mathcal{M}} = \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n} : \boldsymbol{C}^{(k)}\tilde{\boldsymbol{x}} \leq \boldsymbol{0}\right\}$$

Each solution point $\tilde{\boldsymbol{x}}^{(k)}$ in (2.66) is obtained by solving a second-order cone programming (SOCP) problem because set $\widetilde{\mathcal{M}}$ is a convex polytope, set $\widetilde{\mathcal{K}}$ is a closed ball in $\mathbb{R}^{2n}$ under an affine mapping, and the objective function is a linear function of $\tilde{\boldsymbol{x}}$.

Although reformulating the nonsmooth subproblems in (2.59) into the SOCP subproblems in (2.66) entails doubling the dimension of the optimization variable, e.g., from $\boldsymbol{x} \in \mathbb{R}^n$ in (2.44) to $\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n}$ in (2.66), the amount of computation and storage required to solve the SOCP subproblems is *much less* than 2 times that required to solve the nonsmooth subproblems in (2.44). The reduced effort is due to the fact that (1) matrices $\boldsymbol{C}^{(k)}$, $\widetilde{\boldsymbol{A}}$, and vector $\boldsymbol{e}$ in (2.66) are *sparse*, and (2) numerical computing environments such as MATLAB employ data structures to represent sparse vectors and matrices that use storage space proportional to the number of nonzero entries of the sparse data and most of the operations with such sparse data requires time proportional to the number of arithmetic operations on nonzeros entries [49]. For instance, if we let $n_1^{(k)}$, $n_2^{(k)}$, and $n_3^{(k)}$ denote, respectively, the number of coordinates of $\boldsymbol{x}^{(k)}$ in (2.66) such that the conditions $|x_i^{(k)}| \leq \epsilon$, $\epsilon < |x_i^{(k)}| \leq \alpha\epsilon$ and $|x_i^{(k)}| > \alpha\epsilon$ are satisfied, then from (2.65) we conclude that matrix $\boldsymbol{C}^{(k)}$ has only $4(n_1^{(k)} + n_2^{(k)}) + n_3^{(k)}$ nonzero entries. In addition, only half the entries of matrix $\widetilde{\boldsymbol{A}}$ and vector $\boldsymbol{e}$ are nonzero. Hence, the amount of computation and storage required for the solution of the SOCP subproblems is reduced. Moreover, due to the SOCP formulation we can employ standard state-of-the-art SOSs for the solution of SOCP problems such as self-dual-minimization (SeDuMi) [102]. In conclusion, the update formula in (2.66) can be computed efficiently.

2.5 Simulation Results

We now present simulation results to evaluate the proposed and corresponding competing methods in terms of their capability of recovering signals in a wide range of test problems. The proposed and competing methods were evaluated following the experimental protocol discussed in Sec. 1.3. The probability of perfect recovery (PPR) and minimum required fraction (MRF), m/s, for perfect recovery were employed as reconstruction performance (RP) metrics. Perfect recovery was declared when $\nu = 1 \times 10^{-3}$

in (1.36). The average CPU time in seconds was employed as the computation cost (CC) metric. The RP and CC metrics were estimated by carrying out the recovery process 100 times over different sets of measurements. Each measurement vector $\boldsymbol{b}$ was generated by applying a renormalized Gaussian matrix to the signal of interest $\boldsymbol{x}^0$ of length n. The length of vector $\boldsymbol{b}$ was assumed to be $m = n/8$ and the sparsity s of $\boldsymbol{x}^0$ was assumed to be in the range $\frac{m}{100} \leq s \leq \frac{m}{2}$. The MRF, m/s, for perfect recovery was estimated by finding the minimum value of s in that range where PPR $= 1$. The s nonzero values of $\boldsymbol{x}^0$ were chosen randomly from a zero-mean Gaussian distribution of unit variance. In the recovery of noisy signals, each measurement vector $\boldsymbol{b}$ was obtained using a Gaussian vector $\boldsymbol{z}$ with $\sigma_{\boldsymbol{z}} = 1 \times 10^{-4}$. In the case of noiseless signals, we have $\sigma_{\boldsymbol{z}} = 0$ in which case $\boldsymbol{z} = \boldsymbol{0}$.

All experiments were run on a Dell Precision 670 workstation with two 3.2 GHz dual-core Intel Xeon processors and 4 Gb of RAM using the 64-bit Linux MATLAB Version 7.13 (R2011b). Software that is publicly available online was used for the competing methods.[1] We used the values suggested in [11,18,43,68] for the parameters of the competing methods with the exception of the case where such parameter values impact the precision of the solver employed and, therefore, the accuracy of the solution found. The solver precision is directly related to the PPR obtained because solutions that are not accurate result in low PPRs. In such a case, we have increased the solver precision until no further change in the PPR could be observed. We have found out that the PPR of the ℓ_1-Magic and spectral projected-gradient ℓ_1-norm (SPGL1) methods were improved by increasing the default precision of their respective solvers. For instance, the optimality tolerance of the SPGL1 method can be controlled by the parameter "optTol" which has the default value of 1×10^{-4}. In our simulations we have found out that optTol $= 1 \times 10^{-8}$ yields the best results. In the ℓ_1-Magic method, the tolerance for the primal-dual interior-point solver can be controlled by parameter "pdtol" in the sense that the solver terminates if the duality gap is less than or equal to pdtol. The default value of pdtol $= 1 \times 10^{-3}$ has been changed to pdtol $= 1 \times 10^{-8}$

[1] The codes for the competing methods were obtained from the following Web pages:

RLS methods: ℓ_1-LS from S. P. Boyd at http://www.stanford.edu/~boyd/l1_ls/, gradient projection for sparse reconstruction (GPSR) from M. A. T. Figueiredo at http://www.lx.it.pt/~mtf/GPSR/.

Least absolute shrinkage and selection operator (LASSO) methods: SPGL1 from M. P. Friedlander at http://www.cs.ubc.ca/~mpf/spgl1/.

BP methods: ℓ_1-Magic from J. Romberg at http://users.ece.gatech.edu/~justin/l1magic/

to yield the best results. The tolerance for the log-barrier interior-point solver is controlled by parameter "lbtol" and the default value of lbtol $= 1 \times 10^{-3}$ has been changed to lbtol $= 1 \times 10^{-9}$.

2.5.1 RLS Methods

The proposed method of Sec. 2.3 will hereafter be called the QA-SCAD method. The parameters used in the QA-SCAD method were as follows. The initial point was assumed to be $\boldsymbol{x}^{(0)} = \boldsymbol{A}^T\boldsymbol{b}$, convergence was declared when $\varepsilon_c = 1 \times 10^{-7}$, and $\varepsilon_r = 1 \times 10^{-9}$. For the continuation procedure we used a decreasing valued sequence $\{\epsilon_1, \ldots, \varepsilon_5\}$ with $\epsilon_1 = 0.1||\boldsymbol{A}^T\boldsymbol{b}||_\infty$ and $\epsilon_5 = 5 \times 10^{-4}||\boldsymbol{A}^T\boldsymbol{b}||_\infty$. A continuation procedure was also used for the GPSR method with a decreasing valued sequence $\{\lambda_1, \ldots, \lambda_5\}$ where $\lambda_1 = 0.1||\boldsymbol{A}^T\boldsymbol{b}||_\infty$ and $\lambda_5 = 5 \times 10^{-4}||\boldsymbol{A}^T\boldsymbol{b}||_\infty$. The ℓ_1-LS method does not support continuation procedures because the SOS employed is based on an interior-point method. Initialization in interior-point methods is an active research topic [112] and most interior-point methods cannot benefit from an appropriate initialization. Thus, the ℓ_1-LS method was used to solve problem (QP_λ) in (1.8) with $\lambda = 5 \times 10^{-4}||\boldsymbol{A}^T\boldsymbol{b}||_\infty$.

We carried out recovery simulations to evaluate the proposed and competing methods for the recovery of noiseless and noisy signals. The results obtained for noiseless signals of length $n = 512$ and $n = 1,024$ with sparsity $s \leq 5$ and $s \leq 7$, respectively, are summarized in Table 2.1. As can be seen, the QA-SCAD method achieved superior RP relative to those of the GPSR and ℓ_1-LS methods for the case of noiseless signals. The CC of the QA-SCAD method was slightly increased relative to that of the GPSR method and slightly decreased relative to that of the ℓ_1-LS method.

	RP	CC	
n	MRF	CPU time	method
	21.3	0.8062	ℓ_1-LS
512	21.3	0.0424	GPSR
	12.8	0.1714	QA-SCAD
	18.3	2.2706	ℓ_1-LS
$1,024$	32	0.0232	GPSR
	18.3	0.4978	QA-SCAD

Table 2.1: Summary of results for RLS methods and noiseless signals.

The results obtained for noisy signals of length $n = 512$ and $n = 1,024$ with sparsity $s \leq 5$ and $s \leq 7$, respectively, are summarized in Table 2.2. As can be seen, the QA-SCAD method achieved superior RP relative to those of the GPSR and ℓ_1-LS methods for the case of noisy signals. The CC of the QA-SCAD method was increased relative to that of the GPSR method and decreased relative to that of the ℓ_1-LS method.

	RP	CC	
n	MRF	CPU time	method
	21.3	0.8452	ℓ_1-LS
512	21.3	0.0454	GPSR
	12.8	0.2499	QA-SCAD
	18.3	2.5410	ℓ_1-LS
$1,024$	32.0	0.0897	GPSR
	18.3	1.4409	QA-SCAD

Table 2.2: Summary of results for RLS methods and noisy signals.

The results summarized in Tables 2.1 and 2.2 are described in detail in the following subsections.

Results for noiseless signals

RP and CC of the proposed method are compared with those of the ℓ_1-LS and GPSR methods in Fig. 2.4. As can be seen in Figs. 2.4a and 2.4b, the proposed method

achieves superior RP relative to the those of the competing methods for different signal sizes. For instance when $n = 512$, the MRF, m/s, for perfect reconstruction has dropped from $n/24$ in the GPSR and ℓ_1-LS methods to $n/40$ in the QA-SCAD method. On the other hand, when $n = 1,024$ the MRFs for perfect reconstruction of the QA-SCAD and ℓ_1-LS methods were exactly $n/56$ while that of the GPSR method was $n/32$.

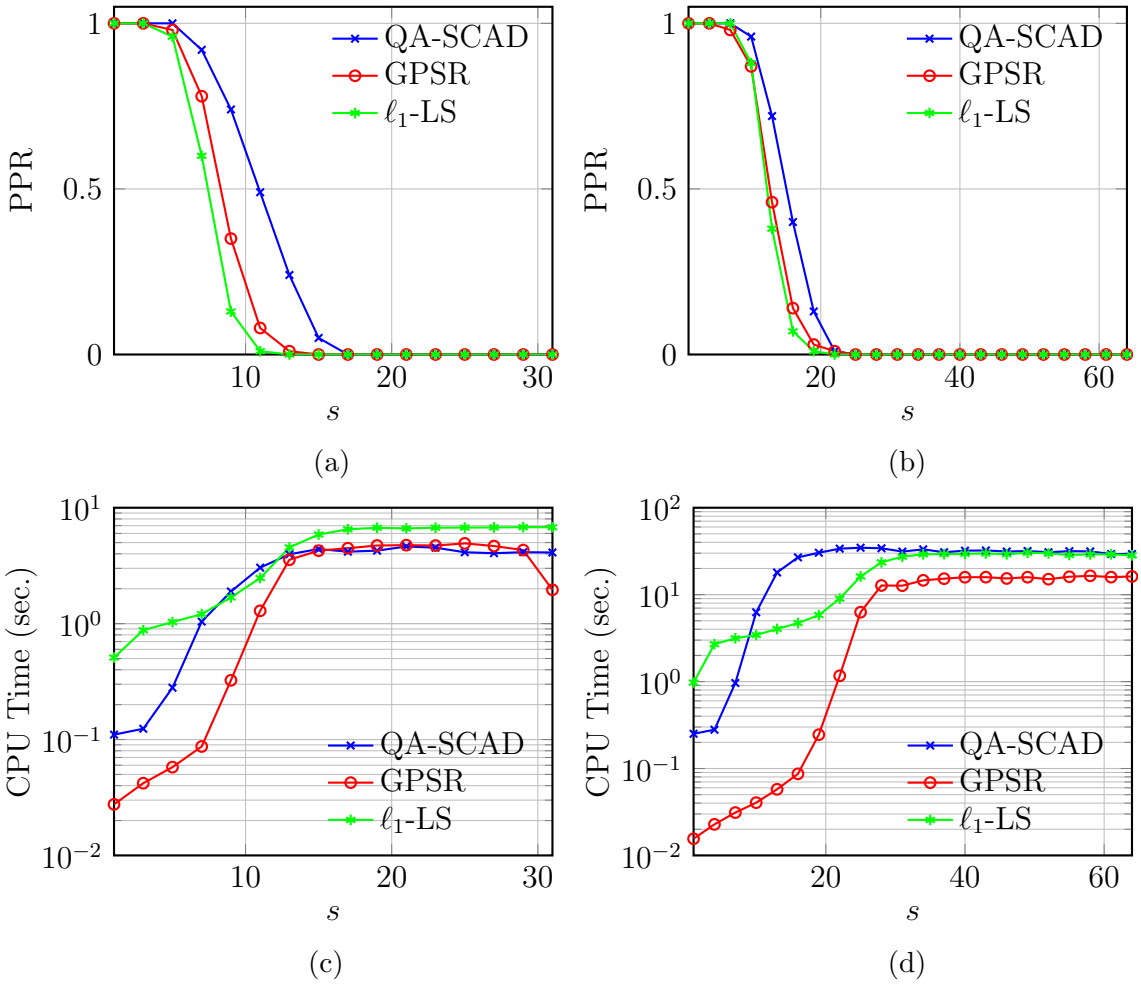


Figure 2.4: RP and CC of RLS-SCAD and competing methods for noiseless signals: (a) PPR for $n = 512$, (b) PPR for $n = 1,024$, (c) Average CPU time for $n = 512$, and (d) Average CPU time for $n = 1,024$.

The RP of the proposed method is also consistently better than those of the competing methods for simulations where the sparse signal was not always perfectly reconstructed. For instance when $n = 512$, the PPRs of the QA-SCAD, GPSR, and ℓ_1-LS methods were 74%, 35%, and 13%, respectively, for $s = 9$. On the other hand,

when $n = 1,024$ the PPRs of the QA-SCAD, GPSR, and ℓ_1-LS methods were 72%, 46%, and 38%, respectively, for $s = 13$.

The CC of the proposed method is comparable to those of the competing methods. As can be seen in Figs. 2.4c and 2.4d, the average CPU time required by the QA-SCAD method is of the same order of magnitude as those required by the ℓ_1-LS and GPSR methods.

The average number of points k computed for sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ before convergence is plotted in Fig. 2.5. As can be seen, a reduced number of points is computed when PPR $= 1$ relative to the number of points computed when PPR $\neq 1$. In addition, for both cases the convergence rate of sequence $\left\{F(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ is improved during the continuation procedure because the average k reduces as the continuation step increases.

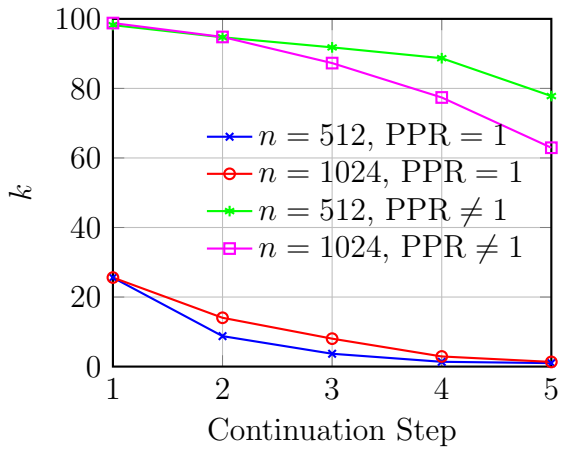


Figure 2.5: Average iteration comparison.

Results for noisy signals

The RP and CC of the proposed method are compared with those of the ℓ_1-LS and GPSR methods in Fig. 2.6. As can be seen in Figs. 2.6a and 2.6b, the proposed method achieves superior RP relative to those of the competing methods for different signal sizes. For instance when $n = 512$, the MRF for perfect reconstruction has dropped from $n/24$ in the GPSR and ℓ_1-LS methods to $n/40$ in the QA-SCAD method. On the other hand, when $n = 1,024$ the MRFs for perfect reconstruction of the QA-SCAD and ℓ_1-LS methods were exactly $n/56$ while that of the GPSR method was $n/32$.

The RP of the proposed method is also consistently better than those of the

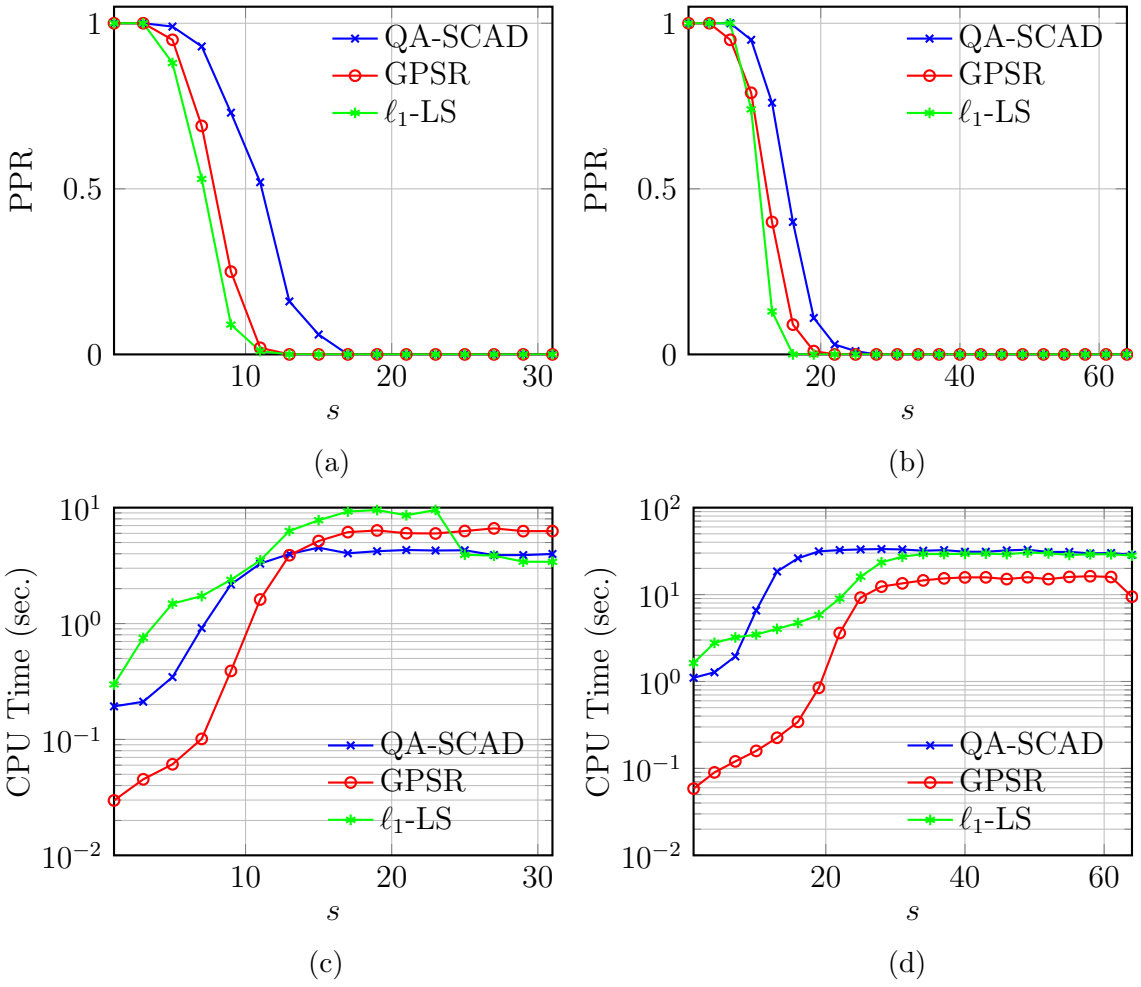


Figure 2.6: RP and CC of RLS-SCAD and competing methods for noisy signals: (a) PPR for $n = 512$, (b) PPR for $n = 1,024$, (c) Average CPU time for $n = 512$, and (d) Average CPU time for $n = 1,024$.

competing methods for simulations where the sparse signal is not always perfectly reconstructed. For instance when $n = 512$, the PPRs of the QA-SCAD, GPSR, and ℓ_1-LS methods were 93%, 69%, and 53%, respectively, for $s = 7$. On the other hand, when $n = 1,024$ the PPR of the QA-SCAD, GPSR, and ℓ_1-LS methods were 76%, 40%, and 13%, respectively, for $s = 13$. The CC of the proposed method and competing methods are compared in Figs. 2.6c and 2.6d. As can be seen, the average CPU time required by the QA-SCAD method is of the same order of magnitude to those required by the ℓ_1-LS and GPSR methods.

The average number of points k computed for sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ before convergence is plotted in Fig. 2.7. As can be seen, a reduced number of points is computed when PPR $= 1$ relative to the number of points computed when PPR $\neq 1$. In addition, for both cases the convergence rate of sequence $\{F(\boldsymbol{x}^{(k)})\}_{k\in\mathbb{N}}$ is improved during the continuation procedure because the average k reduces as the continuation step increases.

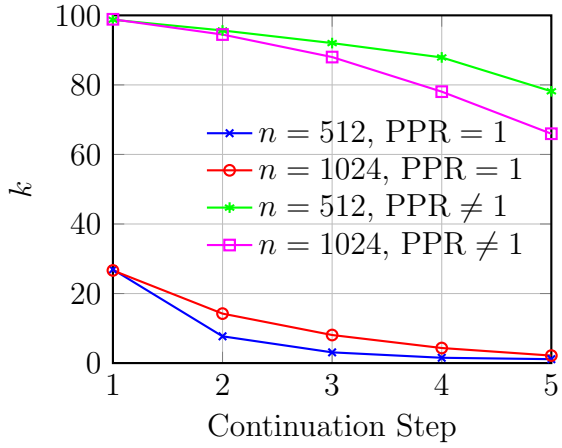


Figure 2.7: Average iteration comparison.

2.5.2 BP Methods

The proposed method of Sec. 2.4 will hereafter be called the PLA-SCAD method. The parameters used in the PLA-SCAD method were as follows. The initial point $\tilde{\boldsymbol{x}}^0 \in \widetilde{\mathcal{K}}$ in (2.66) was defined to be $\tilde{\boldsymbol{x}}^{(0)} = \mathbf{0}$, the regularization parameter ϵ of the SCAD function was set to $\epsilon = 0.1||\boldsymbol{A}^T\boldsymbol{b}||_\infty$, and convergence was declared when $\varepsilon_c = 1 \times 10^{-7}$ in (2.60). We used $\delta = \sigma_{\boldsymbol{z}}\sqrt{m}$ to solve problem (BP$_\delta$) in (1.7).

We carried out recovery simulations to evaluate the proposed and competing methods for the recovery of noiseless and noisy signals. The results obtained for noiseless

signals of length $n = 512$ and $n = 1,024$ with sparsity $s \leq 11$ and $s \leq 22$, respectively, are summarized in Table 2.3. As can be seen, the PLA-SCAD method achieved superior RP relative to those of the ℓ_1-Magic and SPGL1 methods for the case of noiseless signals. The CC of the PLA-SCAD method was increased relative to those of the ℓ_1-Magic and SPGL1 methods.

	RP	CC	
n	MRF	CPU time	method
512	9.14	0.1037	ℓ_1-Magic
	21.33	0.1876	SPGL1
	5.82	4.0050	PLA-SCAD
1,024	6.74	0.6643	ℓ_1-Magic
	18.29	0.5334	SPGL1
	5.82	22.5516	PLA-SCAD

Table 2.3: Summary of results for BP methods and noiseless signals.

The results obtained for noiseless signals of length $n = 512$ and $n = 1,024$ with sparsity $s \leq 9$ and $s \leq 25$, respectively, are summarized in Table 2.4. As can be seen, the PLA-SCAD method achieved superior RP relative to those of the ℓ_1-Magic and SPGL1 methods for the case of noisy signals. The CC of the PLA-SCAD method was increased relative to those of the ℓ_1-Magic and SPGL1 methods.

	RP	CC	
n	MRF	CPU time	method
512	12.8	4.2825	ℓ_1-Magic
	12.8	0.1411	SPGL1
	7.1	9.8821	PLA-SCAD
1,024	12.8	11.0134	ℓ_1-Magic
	9.8	0.5948	SPGL1
	5.1	64.4888	PLA-SCAD

Table 2.4: Summary of results for BP methods and noisy signals.

The results summarized in Tables 2.3 and 2.4 are described in detail in the following subsections.

Results for noiseless signals

RP and CC of the proposed method are compared with those of the ℓ_1-Magic and SPGL1 methods in Fig. 2.8. As can be seen in Figs. 2.8a and 2.8b, the proposed method achieves superior RP relative to those of the competing methods for different signal sizes. For instance when $n = 512$, the MRF for perfect reconstruction has dropped from $n/24$ and $n/56$ in the SGPL1 and ℓ_1-Magic methods, respectively, to $n/88$ in the PLA-SCAD method. On the other hand, when $n = 1,024$ the MRF for perfect reconstruction has dropped from $n/56$ and $n/152$ in the SGPL1 and ℓ_1-Magic methods, respectively, to $n/176$ in the PLA-SCAD method.

The RP of the proposed method is also consistently better than those of the competing methods for simulations where the sparse signal was not always perfectly reconstructed. For instance when $n = 512$, the PPRs of the PLA-SCAD, ℓ_1-Magic, and SPGL1 methods were 80%, 49%, and 24%, respectively, for $s = 13$. On the other hand, when $n = 1,024$ the PPRs of the PLA-SCAD, ℓ_1-Magic, and SPGL1 methods were 82%, 30%, and 5%, respectively, for $s = 28$. However, superior RP of the proposed method come with an additional CC. As can be seen in Figs. 2.8c and 2.8d, the average CPU time required by the PLA-SCAD method is increased relative to those of the ℓ_1-Magic and SPGL1 methods.

The average number of points k computed for sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ before convergence is plotted in Fig. 2.9. When the signal is always perfectly reconstructed, i.e., when $s \leq 11$ and $s \leq 22$ in Figs. 2.9a and 2.9b, respectively, a reduced number of points is computed. In such a case, 2 points are computed on average before convergence. However, when $11 < s < 21$ and $22 < s < 37$ in Figs. 2.9a and 2.9b, respectively, the average number of points computed increases for increasing values of s. On the other hand, this increase ceases when $s > 21$ and $s > 37$ in Figs. 2.9a and 2.9b, respectively. In such a case, 17 and 25 points are computed on average before convergence.

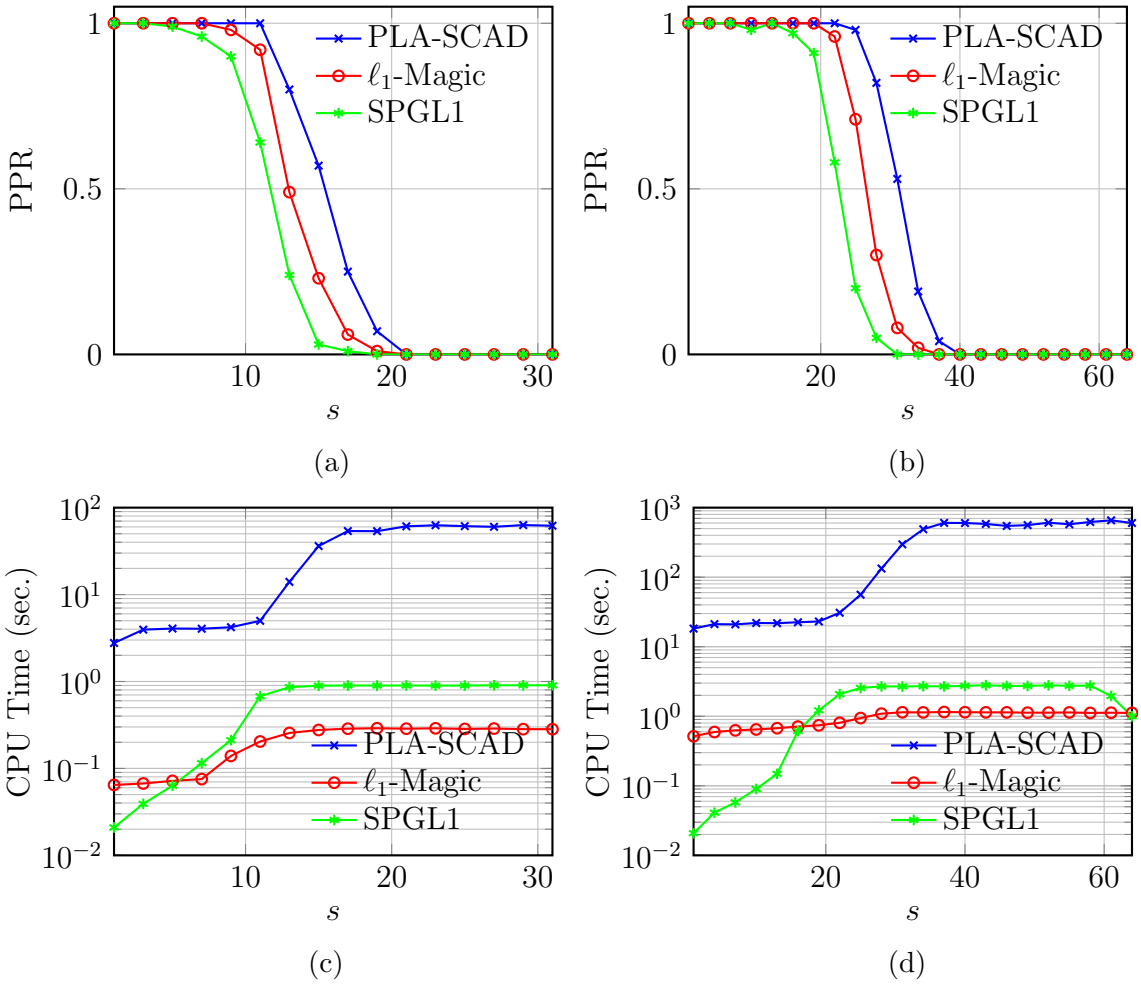


Figure 2.8: RP and CC of BP-SCAD and competing methods for noiseless signals: (a) PPR for $n = 512$, (b) PPR for $n = 1,024$, (c) Average CPU time for $n = 512$, and (d) Average CPU time for $n = 1,024$.

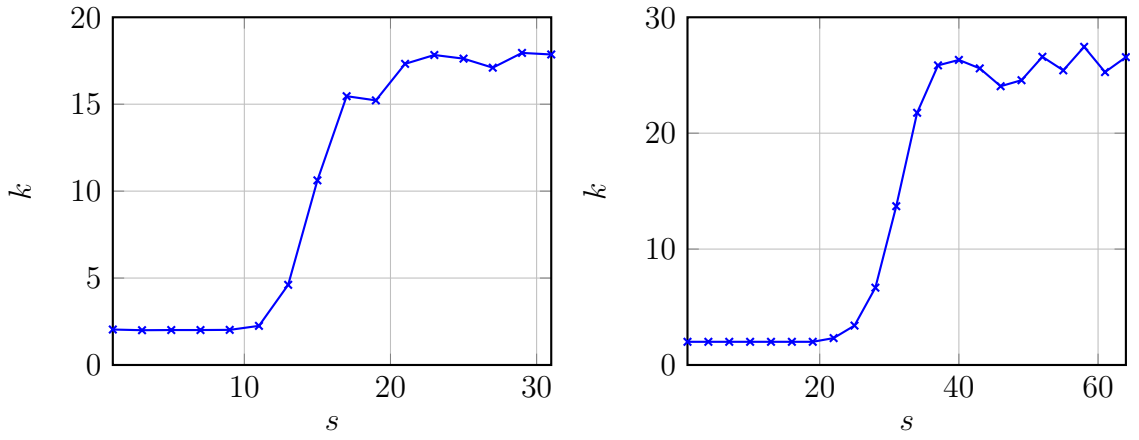


Figure 2.9: Average iteration comparison: (a) $n = 512$ and (b) $n = 1,024$.

Results for noisy signals

RP and CC of the proposed method are compared with those of the ℓ_1-Magic and SPGL1 methods in Fig. 2.10. As can be seen in Figs. 2.10a and 2.10b, the proposed method achieves superior RP relative to those of the competing methods for different signal sizes. For instance when $n = 512$, the MRF for perfect reconstruction has dropped from $n/40$ in the SGPL1 and ℓ_1-Magic methods to $n/72$ in the PLA-SCAD method. On the other hand, when $n = 1,024$ the MRF for perfect reconstruction has dropped from $n/80$ and $n/104$ in the ℓ_1-Magic and SGPL1 methods, respectively, to $n/200$ in the PLA-SCAD method.

The RP of the proposed method is also consistently better than those of the competing methods for simulations where the sparse signal was not always perfectly reconstructed. For instance when $n = 512$, the PPRs of the PLA-SCAD, ℓ_1-Magic, and SPGL1 methods were 85%, 25%, and 24%, respectively, for $s = 13$. On the other hand, when $n = 1,024$ the PPRs of the PLA-SCAD, SPGL1, and ℓ_1-Magic were 77%, 2%, and 1%, respectively, for $s = 28$. However, superior RP of the proposed method come with an additional CC. As can be seen in Figs. 2.10c and 2.10d, the average CPU time required by the PLA-SCAD method is increased relative to those of the ℓ_1-Magic and SPGL1 methods.

The average number of points k computed for sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ before convergence is plotted in Fig. 2.11. When the signal is always perfectly reconstructed, i.e., when $s \leq 9$ and $s \leq 25$ in Figs. 2.11a and 2.11b, respectively, a reduced number of

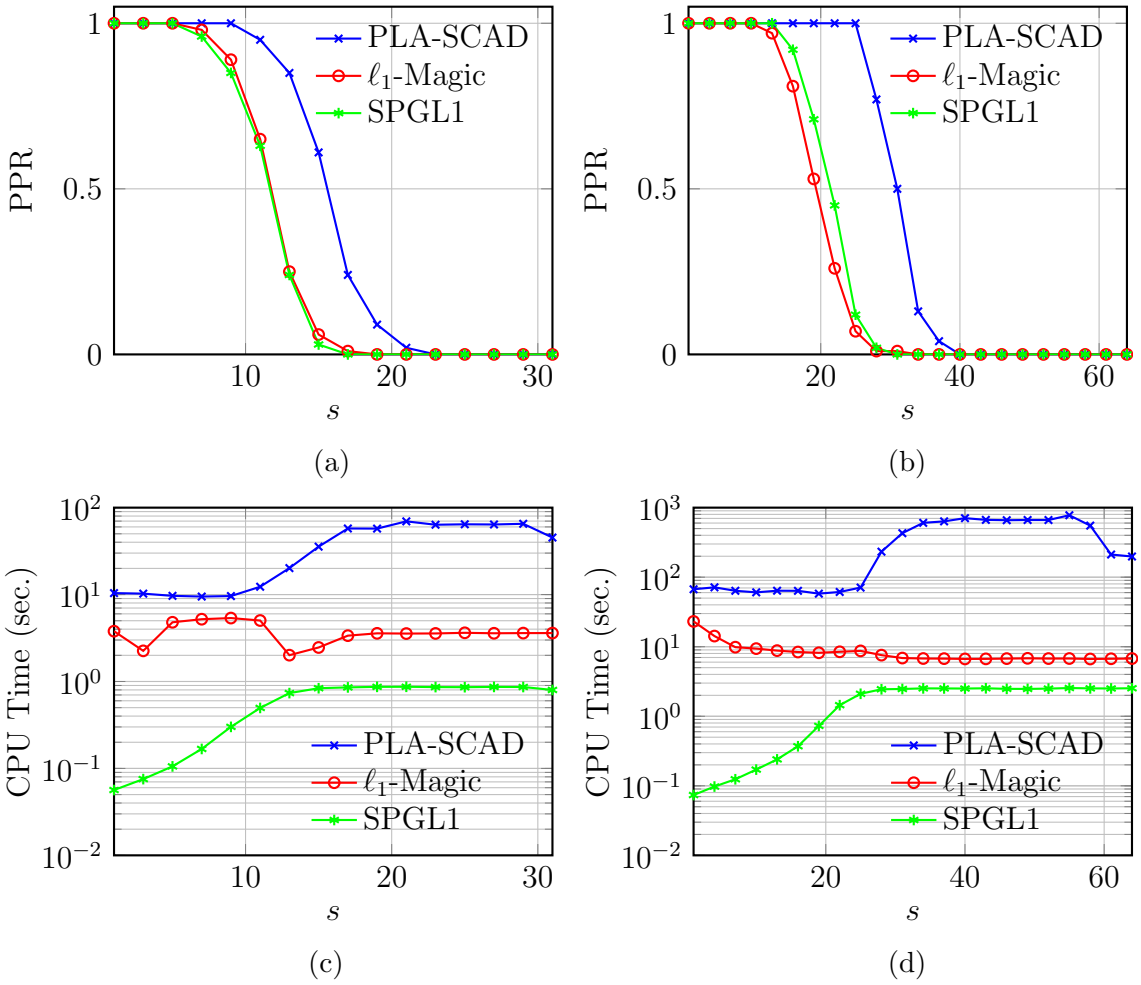
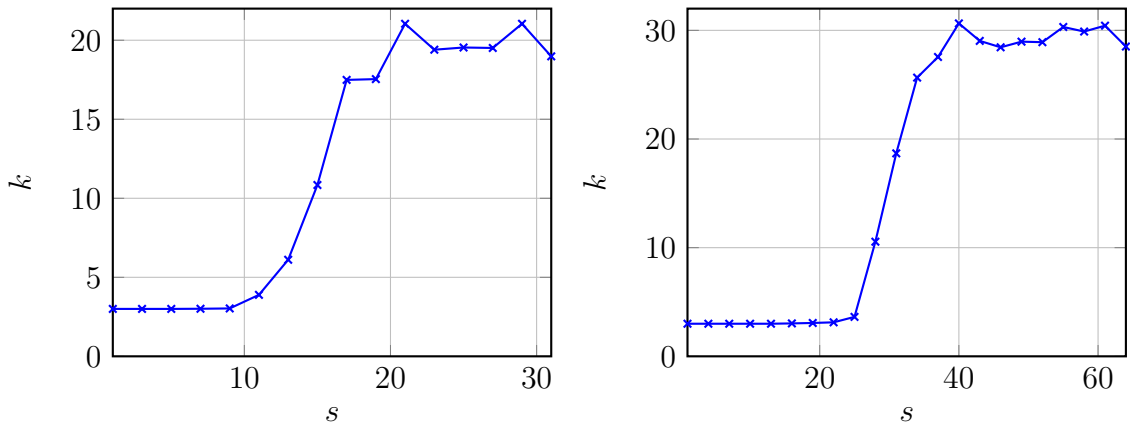


Figure 2.10: RP and CC of BP-SCAD and competing methods for noisy signals: (a) PPR for $n = 512$, (b) PPR for $n = 1,024$, (c) Average CPU time for $n = 512$, and (d) Average CPU time for $n = 1,024$.

points is computed. In such a case, 3 points are computed on average before convergence. However, when $9 < s < 21$ and $25 < s < 37$ in Figs. 2.11a and 2.11b, respectively, the average number of points computed increases for increasing values of s. On the other hand, this increase ceases when $s > 21$ and $s > 37$ in Figs. 2.11a and 2.11b, respectively. In such a case, 19 and 28 points are computed on average before convergence.



(a) (b)

Figure 2.11: Average iteration comparison: (a) $n = 512$ and (b) $n = 1,024$.

2.6 Conclusions

Two new SCF methods that are applicable for the recovery of sparse signals from Gaussian ensembles have been described [104,105]. Sparsity is promoted by using the SCAD function which is known to satisfy certain conditions for unbiasedness, sparsity, and continuity of the solution of the recovery problem. Convex approximations of the SCAD function such as the QA and the PLA have been presented to render the computation of the local minimizer tractable and several results pertaining to the applicability of the approximations to SCF methods were obtained.

In the proposed QA-SCAD method, a solution of problem (QP_λ) in (1.8) is approached by employing the QA of the SCAD function [104]. Convex subproblems are solved by using an SOS where the Newton step can be computed efficiently. A target value of the regularization term of the recovery problem is approached efficiently by using a continuation procedure. In the proposed PLA-SCAD method, a solution to problem (BP_δ) in (1.7) is approached by employing a PLA of the SCAD function [105].

Convex subproblems are reformulated as SOCP problems and are solved efficiently by standard SOSs such as SeDuMi [102].

Simulation results demonstrated that the proposed QA-SCAD and PLA-SCAD methods achieve superior RP metrics in terms of increased PPRs and reduced MRFs for perfect recovery when compared with competing RLS and BP methods, respectively. The CC metrics of the proposed QA-SCAD method was found to be comparable to those of corresponding competing methods, namely, the GPSR [43] and ℓ_1-LS [68] methods. On the other hand, the CC metric of the proposed PLA-SCAD method is increased relative to those of corresponding competing methods, namely, ℓ_1-Magic [18] and SPGL1 [11] methods.

Chapter 3

A New Family of Sequential Convex Formulation Methods

3.1 Introduction

The use of nonconvex sparsity-promoting functions (SPFs) that closely resemble the ℓ_0-norm function is desirable in compressive-sensing (CS) recovery problems because they lead to short signal representations and small reconstruction error [23, 24, 26, 27, 37, 45, 46, 97, 104, 105, 109]. However, state-of-the-art iterative methods for such recovery problems are typically inefficient and lack convergence analysis (see Sec. 1.4 for details).

In this chapter, a new family of sequential convex formulation (SCF) methods that solve nonconvex optimization problems is proposed. The new methods are suitable for large-scale recovery problems. Sparsity is promoted with a fairly general class of nonconvex SPFs that include widely used SPFs as special cases. Optimality conditions are obtained and local minimizers and saddle points of the problem are identified. The new family of methods is based on the update formula in (1.22) and a piecewise-linear approximation (PLA) of the SPF is employed to render computation of the minimizer tractable. Subproblems of the type in (1.22) are formulated as weighted ℓ_1-norm minimization problems while an efficient first-order solver (FOS) suitable for the recovery of large signals from Gaussian or orthogonal ensembles is employed. The sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ in (1.22) is shown to be a monotonically decreasing sequence of values of the objective function and, consequently, converges to a local minimizer. Simulation results demonstrate that the new methods are robust, lead to

fast convergence, and yield solutions that are superior to those achieved with some competing state-of-the-art methods.

The chapter is organized as follows. In Sec. 3.2, a class of nonconvex recovery problems is proposed and its optimality conditions are examined. In Sec. 3.3, a new family of SCF methods is described. In Sec. 3.4, simulation results for the proposed and corresponding competing methods are presented. In Sec. 3.5, conclusions are drawn.

3.2 Proposed Class of Recovery Problems

This dissertation is concerned with recovery problems based on SPFs that share some common mathematical properties such as concavity, differentiability, and boundedness. It is, therefore, convenient to present a family of SPFs in terms of the following definition.

Definition 3.1. *Let $z \in \mathbb{R}_0^+$ where $\mathbb{R}_0^+$ denotes the set of all nonnegative real numbers and let $\mathcal{P}$ denote the set of all SPFs with properties:*

1. *$p_\epsilon(z)$ is second-order continuously differentiable,*
2. *$p_\epsilon(z)$ has a bounded second-order derivative,*
3. *$p_\epsilon(z)$ is concave,*
4. *$p_\epsilon(z)$ is nondecreasing for increasing values of z, and*
5. *$p_\epsilon(z)$ is bounded from below by a real number A*

and there exists a constant $\epsilon' > 0$ such that $A \geq 0$ for $\epsilon \geq \epsilon'$.

Class $\mathcal{P}$ is of practical interest because (1) it is a subset of class $\mathcal{N}$ that includes several contemporary SPF members identified in 1.1.1, and (2) it leads to an efficient and robust recovery process. Thus, optimality conditions and efficient methods for a wide range of nonconvex recovery problems can be obtained. Some examples are described below.

The SPF in (1.4) is a member of $\mathcal{P}$ because function $p_\epsilon(z) = (z+\epsilon)^p$ satisfies Properties 1 to 5, namely, function $p_\epsilon(z)$ is a second-order continuously differentiable function with a second-order derivative given by

$$p''_\epsilon(z) = p(p-1)(z+\epsilon)^{p-2}$$

and $p''_\epsilon(z)$ is bounded because $|p''_\epsilon(z)| \leq p|p-1|\epsilon^{p-2}$. Function $p_\epsilon(z)$ is (1) concave because $p''_\epsilon(z)$ can assume only negative values, (2) nondecreasing for increasing values of z (see Fig. 1.1), and (3) bounded from below by ϵ^p because $p_\epsilon(z) \geq \epsilon^p$ for all $z \in \mathbb{R}_0^+$.

A second example of an SPF member of $\mathcal{P}$ is given in (1.5). Function $p_\epsilon(z) = \ln(z+\epsilon)$ is a second-order continuously differentiable function with a second-order derivative given by

$$p''_\epsilon(z) = -1/(z+\epsilon)^2$$

and $p''(z)$ is bounded because $|p''_\epsilon(z)| \leq 1/\epsilon^2$. Function $p_\epsilon(z)$ is (1) concave because $p''_\epsilon(z)$ can assume only negative values, (2) nondecreasing for increasing values of z (see Fig. 1.2), and (3) bounded from below by $\ln(\epsilon)$ because $p_\epsilon(z) \geq \ln(\epsilon)$ for all $z \in \mathbb{R}_0^+$. Furthermore, $\ln(\epsilon) \geq 0$ for $\epsilon \geq 1$.

Yet another example of an SPF member of $\mathcal{P}$ is given in (1.6). Function $p_\epsilon(z)$ given by

$$p_\epsilon(z) = \begin{cases} \epsilon z, & z \leq \epsilon \\ -\left[z^2 - 2\alpha\epsilon z + \epsilon^2\right] / \left[2(\alpha-1)\right], & \epsilon < z \leq \alpha\epsilon \\ (\alpha+1)\epsilon^2/2, & z > \alpha\epsilon \end{cases}$$

is a second-order continuously differentiable function with a second-order derivative given by

$$p''_\epsilon(z) = \begin{cases} -1/(\alpha-1), & \epsilon < z < \alpha\epsilon \\ 0, & \text{otherwise} \end{cases}$$

and $p''_\epsilon(z)$ is bounded because $|p''_\epsilon(z)| \leq 1/(\alpha-1)$. Function $p_\epsilon(z)$ is (1) concave because $p''(z)$ can assume only nonpositive values, (2) nondecreasing for increasing values of z (see Fig. 1.3), and (3) bounded from below by 0 because $p_\epsilon(z) \geq 0$ for all $z \in \mathbb{R}_0^+$.

On the other hand, an example of an SPF member of $\mathcal{N}$ that is not included in $\mathcal{P}$ is given by

$$p_\epsilon(|x_i|) = |x_i|^p \tag{3.1}$$

where $0 < p < 1$. Here function $P_\epsilon(\boldsymbol{x})$ is equivalent to the ℓ_p^p norm of $\boldsymbol{x}$. The SPF in (3.1) is not included in $\mathcal{P}$ because the second-order derivative of $p_\epsilon(z) = z^p$ given by

$$p''_\epsilon(z) = p(p-1)z^{p-2}$$

is undefined at $z = 0$.

Reconstruction problems where signal recovery is performed by solving problem (BP_δ) in (1.7) with $p_\epsilon(|x_i|) \in \mathcal{P}$ are referred to as $\mathcal{P}$-class problems hereafter. In Sec. 3.2.1, an equivalent formulation of $\mathcal{P}$-class problems facilitating the analysis of optimality conditions is introduced. In Sec. 3.2.2, optimality conditions on $\mathcal{P}$-class problems are presented and minimizers are identified.

3.2.1 Smooth reformulation

A $\mathcal{P}$-class problem is a nonsmooth optimization problem because $p_\epsilon(|x_i|) \in \mathcal{P}$ is a nonsmooth function of x_i. Nonsmoothness can be circumvented by applying a variable transformation to the argument of the absolute value function as specified in p. 148 of [44]. Let $\mathcal{I}_n$ and $\mathcal{J}_n$ denote sets of n integers and $\widetilde{\mathcal{I}}_{2n}$ denote a set of $2n$ integers given by

$$\mathcal{I}_n = \{1, \ldots, n\}\,, \quad \mathcal{J}_n = \{n+1, \ldots, 2n\}\,, \text{ and } \quad \widetilde{\mathcal{I}}_{2n} = \mathcal{I}_n \cup \mathcal{J}_n \tag{3.2}$$

The variable transformation is accomplished by introducing variables $u_i = \max(x_i, 0)$ and $v_i = \max(-x_i, 0)$, and by letting $x_i = u_i - v_i$ and $|x_i| = u_i + v_i$ in (1.7). Thus, $\mathcal{P}$-class problems can be expressed in the following equivalent smooth formulation

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{v}}{\text{minimize}} \quad & P_\epsilon(\boldsymbol{u}, \boldsymbol{v}) \\ \text{subject to:} \quad & \|\boldsymbol{A}(\boldsymbol{u} - \boldsymbol{v}) - \boldsymbol{b}\|_2 \leq \delta \\ & u_i \geq 0,\ v_i \geq 0, \quad i \in \mathcal{I}_n \end{aligned} \tag{3.3}$$

where

$$P_\epsilon(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i \in \mathcal{I}_n} p_\epsilon(u_i + v_i) \tag{3.4}$$

Introducing optimization variable $\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n}$ given by

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} \tag{3.5}$$

we can write function $P_\epsilon(\boldsymbol{u}, \boldsymbol{v})$ in (3.4) as

$$P_\epsilon(\tilde{\boldsymbol{x}}) = \sum_{i \in \mathcal{I}_n} p_\epsilon(\tilde{x}_i + \tilde{x}_{i+n}) \tag{3.6}$$

The problem in (3.3) can now be rewritten in the compact form

$$\underset{\tilde{\boldsymbol{x}}\in\widetilde{\mathcal{C}}}{\text{minimize}} \quad P_\epsilon(\tilde{\boldsymbol{x}}) \tag{3.7}$$

where $\widetilde{\mathcal{C}}$ is the *convex* feasible set given by

$$\widetilde{\mathcal{C}} = \widetilde{\mathcal{K}} \cap \widetilde{\mathcal{M}} \tag{3.8}$$

Set $\widetilde{\mathcal{K}}$ in (3.8) is the closed ball in $\mathbb{R}^{2n}$ under an affine mapping given by

$$\widetilde{\mathcal{K}} = \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n} : ||\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} - \boldsymbol{b}||_2 \leq \delta\right\} \tag{3.9}$$

where $\widetilde{\boldsymbol{A}} = \begin{bmatrix}\boldsymbol{A} & -\boldsymbol{A}\end{bmatrix}$, and set $\widetilde{\mathcal{M}}$ in (3.8) is of the form

$$\widetilde{\mathcal{M}} = \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n} : \tilde{\boldsymbol{x}} \geq \boldsymbol{0}\right\} \tag{3.10}$$

In conclusion, nonsmooth $\mathcal{P}$-class problems can be expressed as smooth problems with the formulation in (3.7). This formulation entails doubling the dimension of the optimization variable, i.e., from $\boldsymbol{x} \in \mathbb{R}^n$ to $\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n}$. The increase in problem dimension is of no concern as the reformulation is used for analysis purposes only.

3.2.2 Optimality conditions

Consider the closely related optimization problem

$$\underset{\tilde{\boldsymbol{x}}\in\widetilde{\mathcal{D}}}{\text{minimize}} \quad P_\epsilon(\tilde{\boldsymbol{x}}) \tag{3.11}$$

where $P_\epsilon(\tilde{\boldsymbol{x}})$ is given by (3.6) and $\widetilde{\mathcal{D}}$ is the *convex* feasible set defined by

$$\widetilde{\mathcal{D}} = \widetilde{\mathcal{H}} \cap \widetilde{\mathcal{M}} \tag{3.12}$$

Set $\widetilde{\mathcal{H}}$ in (3.12) is the solution set of a linear system of equations of the form

$$\widetilde{\mathcal{H}} = \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n} : \widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = \boldsymbol{c}\right\} \tag{3.13}$$

where vector $\boldsymbol{c}$ denotes an arbitrary column vector of length m. Set $\widetilde{\mathcal{M}}$ in (3.12) and matrix $\widetilde{\boldsymbol{A}}$ in (3.13) are identical to those defined for the problem in (3.7).

A point $\tilde{\boldsymbol{x}}_*$ is a stationary point of the problem in (3.11) if it satisfies the Karush-Kuhn-Tucker (KKT) conditions

$$\boldsymbol{\nabla} P_\epsilon(\tilde{\boldsymbol{x}}_*) + \widetilde{\boldsymbol{A}}^T \boldsymbol{\lambda}_* + \boldsymbol{\mu}_* = \boldsymbol{0} \tag{3.14a}$$

$$\widetilde{\boldsymbol{A}} \tilde{\boldsymbol{x}}_* = \boldsymbol{c} \tag{3.14b}$$

$$\tilde{\boldsymbol{x}}_* \geq \boldsymbol{0} \tag{3.14c}$$

$$\boldsymbol{\mu}_* \geq \boldsymbol{0} \tag{3.14d}$$

$$\tilde{x}_{i*} \mu_{i*} = 0 \qquad \text{for } i \in \widetilde{\mathcal{I}}_{2n} \tag{3.14e}$$

where column vectors $\boldsymbol{\lambda}_*$ and $\boldsymbol{\mu}_*$ of length m denote the optimal Lagrange multipliers (see Section 7.2 of [14]).

We identify three types of stationary points of the problem in (3.11) that are of interest for our analysis, namely, global and local minimizers and saddle points. A global minimizer of the problem in (3.11) is a stationary point where function $P_\epsilon(\tilde{\boldsymbol{x}})$ attains its minimum value over feasible set $\widetilde{\mathcal{D}}$, as stated in the following definition.

Definition 3.2. *Stationary point $\tilde{\boldsymbol{x}}_*$ is a global minimizer of the problem in (3.11) if $P_\epsilon(\tilde{\boldsymbol{x}}) \geq P_\epsilon(\tilde{\boldsymbol{x}}_*)$ for all $\tilde{\boldsymbol{x}} \in \widetilde{\mathcal{D}}$.*

A local minimizer of the problem in (3.11) is a stationary point where function $P_\epsilon(\tilde{\boldsymbol{x}})$ attains its minimum value within a given neighborhood. In other words, by considering a ball of radius $\alpha > 0$ centered at point $\tilde{\boldsymbol{x}}_*$, or equivalently an α-neighborhood around $\tilde{\boldsymbol{x}}_*$ given by

$$B(\tilde{\boldsymbol{x}}_*, \alpha) = \{\tilde{\boldsymbol{x}} : \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}_*\|_2 \leq \alpha\} \tag{3.15}$$

we define the local minimizer as follows.

Definition 3.3. *Stationary point $\tilde{\boldsymbol{x}}_*$ is a local minimizer of the problem in (3.11) if there exists an $\alpha > 0$ such that $P_\epsilon(\tilde{\boldsymbol{x}}) \geq P_\epsilon(\tilde{\boldsymbol{x}}_*)$ for any $\tilde{\boldsymbol{x}} \in B(\tilde{\boldsymbol{x}}_*, \alpha) \cap \widetilde{\mathcal{D}}$.*

On the other hand, a saddle point of the problem in (3.11) is a stationary point in which $P_\epsilon(\tilde{\boldsymbol{x}}_*)$ is neither a minimum nor a maximum within the neighborhood given by (3.15).

It is well known that the KKT conditions are sufficient for optimality in convex minimization problems (see Proposition 2.1.2 of [14]). For instance, if we suppose

that function $P_\epsilon(\tilde{\boldsymbol{x}})$ is convex, then a stationary point $\tilde{\boldsymbol{x}}_*$ satisfying the conditions in (3.14) must be a global minimizer of the problem in (3.11). Unfortunately, such a convexity assumption is not valid for the problem at hand because function $P_\epsilon(\tilde{\boldsymbol{x}})$ is based on an SPF of class $\mathcal{P}$ and is, therefore, concave. The problem of minimizing a concave function over a convex set, as defined in (3.11), is equivalent to the problem of maximizing a convex function over a convex set [59], i.e., maximization and minimization problems are related by

$$\max_{\tilde{\boldsymbol{x}}\in\widetilde{\mathcal{D}}} \bar{P}_\epsilon(\tilde{\boldsymbol{x}}) = -\min_{\tilde{\boldsymbol{x}}\in\widetilde{\mathcal{D}}} [P_\epsilon(\tilde{\boldsymbol{x}})]$$

where $\bar{P}_\epsilon(\tilde{\boldsymbol{x}}) = -P_\epsilon(\tilde{\boldsymbol{x}})$, and thus $\bar{P}_\epsilon(\tilde{\boldsymbol{x}})$ is convex. A well known result pertaining to the optimality of convex maximization problems shows that the KKT conditions are only necessary for optimality (see Theorem 32.4 of [95]). Thus, a stationary point $\tilde{\boldsymbol{x}}_*$ satisfying the conditions in (3.14) may not be a global minimizer of the problem in (3.11), e.g., it may be a local minimizer or a saddle point. We can identify a stationary point of the problem in (3.11) as a global minimizer, local minimizer, or saddle point by looking at the number of nonzero coordinates of such a point.

Let $\widetilde{\mathcal{S}}$ denote the set of all stationary points of the problem in (3.11), and let $\widetilde{\mathcal{S}}^r$ denote the subset of all points in $\widetilde{\mathcal{S}}$ that have precisely r nonzero coordinates. Furthermore, let $\tilde{\boldsymbol{x}}_*^r$ represent a member of $\widetilde{\mathcal{S}}^r$. First, we identify $\tilde{\boldsymbol{x}}_*^r$ as a global minimizer by using the concept of *extreme point* of a convex set. The feasible set in (3.12) can be written as $\widetilde{\mathcal{D}} = \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{2n} : \widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = \boldsymbol{c},\ \tilde{\boldsymbol{x}} \geq \boldsymbol{0}\right\}$ which shows that it assumes the form of a *polytope*, i.e., a convex and connected set with flat, polygonal faces (see p. 356 of [85]). Several properties of the members of this polytope can be deduced for the case where matrix $\widetilde{\boldsymbol{A}}$ is of full row rank. We have a result pertaining to the rank of sensing matrix $\boldsymbol{A}$ and, consequently, to the rank of $\widetilde{\boldsymbol{A}}$, as described in the following proposition.

Proposition 3.1 (Full Row-Rank Sensing Matrix)**.** *Sensing matrix* $\boldsymbol{A}$ *is of* full row rank*, i.e.,* $\operatorname{rank}(\boldsymbol{A}) = m$*.*

Proof. Consider the case of Gaussian ensembles and let $\bar{\boldsymbol{W}} = \bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}}$ with $\bar{\boldsymbol{A}} = \boldsymbol{A}^T$. Since $m \leq n$, from Theorem 3.1.4 of [79] we obtain

$$\bar{\boldsymbol{W}} \succ 0 \quad \text{and} \quad \operatorname{rank}\left(\bar{\boldsymbol{W}}\right) = m \tag{3.16}$$

with probability 1. By using the properties of the rank of a matrix [58], we get

$$\text{rank}\left(\bar{\boldsymbol{W}}\right)=\text{rank}\left(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}}\right)=\text{rank}\left(\bar{\boldsymbol{A}}\right)=\text{rank}\left(\bar{\boldsymbol{A}}^T\right) \tag{3.17}$$

and since $\bar{\boldsymbol{A}}^T = \boldsymbol{A}$, from (3.17), we must have

$$\text{rank}\left(\bar{\boldsymbol{A}}^T\right) = \text{rank}\left(\boldsymbol{A}\right) \tag{3.18}$$

Hence, combining (3.16) to (3.18), we obtain $\text{rank}\left(\boldsymbol{A}\right) = m$ with probability 1.

For the case of orthogonal ensembles, (1) the rows of $\boldsymbol{A}$ are orthogonal and, therefore, linearly independent, and (2) dimensions of $\boldsymbol{A}$ are such that $m \leq n$. Thus, $\text{rank}\left(\boldsymbol{A}\right) = m$. □

For a polytope defined by a full row-rank matrix, such as that in (3.12), members with r nonzero coordinates are extreme points of the polytope when $r \leq m$ (see Sec. 13.2 of [85]). Furthermore, the global minimum of a concave function relative to a convex set occurs at some extreme point of the set (see p. 342 of [95]). Based on these results, a global minimizer of the problem in (3.11) is identified in the following lemma.

Lemma 3.1 (Global Minimizer of Problem in (3.11))**.** *A global minimizer of the problem in* (3.11) *is a member of set* $\widetilde{\mathcal{S}}^r$ *when* $r \leq m$.

Proof. We note that (1) stationary point $\tilde{\boldsymbol{x}}_*^r$ must be a member of polytope $\widetilde{\mathcal{D}}$ on the basis of the KKT conditions in (3.14b) and (3.14c), and (2) polytope $\widetilde{\mathcal{D}}$ is defined by a full-row rank matrix on the basis of Proposition 3.1. Therefore, members of $\widetilde{\mathcal{S}}^r$ with $r \leq m$ are extreme points (or vertices) of the polytope while members of $\widetilde{\mathcal{S}}^r$ with $r > n$ are not (see Theorem 13.3 of [85]). Thus, a stationary point $\tilde{\boldsymbol{x}}_*^r$ with $r \leq m$ is an extreme point of polytope $\widetilde{\mathcal{D}}$. Furthermore, the global minimum of $P_\epsilon(\tilde{\boldsymbol{x}})$ relative to $\widetilde{\mathcal{D}}$ is attained at an extreme point of $\widetilde{\mathcal{D}}$ (see Corollary 32.3.1 of [95]).

In summary, (1) $\tilde{\boldsymbol{x}}_*^r$ with $r \leq m$ is an extreme point of $\widetilde{\mathcal{D}}$, (2) a global minimizer of the problem in (3.11) is an extreme point of $\widetilde{\mathcal{D}}$, and (3) $\tilde{\boldsymbol{x}}_*^r$ satisfies the KKT conditions which are necessary for the optimality of the problem in (3.11). Therefore, a global minimizer of the problem in (3.11) is a member of $\widetilde{\mathcal{S}}^r$ when $r \leq m$. □

We note that a global minimizer is also a local minimizer but the converse is not necessarily true, see Definitions 3.2 and 3.3. As stated in Lemma 3.1, a global

minimizer is a member of $\widetilde{\mathcal{S}}^r$ when $r \leq m$. In fact, every member of $\widetilde{\mathcal{S}}^r$ with $r \leq m$ is a local minimizer as described in the following lemma.

Lemma 3.2 (Local Minimizers of Problem in (3.11))**.** *The points in set* $\widetilde{\mathcal{S}}^r$ *with* $r \leq m$ *are* local *minimizers of the problem in* (3.11).

Proof. A point $\tilde{\boldsymbol{x}}_*^r \in \widetilde{\mathcal{S}}^r$ with $r \leq m$ is an extreme point of polytope $\widetilde{\mathcal{D}}$ and; without loss of generality, let the r nonzero-valued coordinates of $\tilde{\boldsymbol{x}}_*^r$ be $\tilde{x}_1,\ \tilde{x}_2,\ \ldots, \tilde{x}_r$ and the $(2n-r)$ zero-valued coordinates $\tilde{x}_{r+1},\ \tilde{x}_{r+2},\ \ldots, \tilde{x}_{2n}$. For a feasible point $\tilde{\boldsymbol{x}} \in \widetilde{\mathcal{D}}$, let

$$\widetilde{\mathcal{F}} = \left\{ \tilde{\boldsymbol{d}} \in \mathbb{R}^{2n} : \tilde{\boldsymbol{x}} + \alpha \tilde{\boldsymbol{d}} \in \widetilde{\mathcal{D}} \right\} \quad \forall \alpha \in (0, \alpha'), \quad \alpha' > 0$$

denote the cone of the feasible directions of $\widetilde{\mathcal{D}}$ where each vector $\tilde{\boldsymbol{d}} \in \widetilde{\mathcal{F}}$ is a feasible direction (see Definition 4.2.1 of [7]). An edge of polytope $\widetilde{\mathcal{D}}$ is a line segment which connects adjacent extreme points. Members of cone $\widetilde{\mathcal{F}}$ that increase one of the r zero-valued coordinates of an extreme point while keeping all the remaining $(2n-r-1)$ fixed at zero are edge directions from that extreme point to an adjacent extreme point (see pp. 8 and 14 of [50]). Now consider the edge direction

$$\tilde{\boldsymbol{d}}^{r+1} = \begin{bmatrix} d_1 & d_2 & \cdots & d_{r+1} & 0 & \cdots & 0 \end{bmatrix}^T \quad \text{with} \quad ||\tilde{\boldsymbol{d}}^{r+1}||_2 = 1$$

Since $\tilde{\boldsymbol{d}}^{r+1} \in \widetilde{\mathcal{F}}$, there exists an appropriate and fixed $\alpha_{m+1} > 0$ such that $\tilde{\boldsymbol{x}}_*^r + \alpha \tilde{\boldsymbol{d}}^{r+1}$ is feasible for any $\alpha \in (0, \alpha_{m+1})$, which implies that $d_{r+1} > 0$. Furthermore, $\tilde{\boldsymbol{d}} \neq \boldsymbol{0}$ is a member of cone $\widetilde{\mathcal{F}}$ if and only if $\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{d}} = \boldsymbol{0}$ and $\tilde{\boldsymbol{d}} \geq \boldsymbol{0}$ (see Definition 2.13 of [50]), which implies that $d_1,\ d_2,\ \ldots,\ d_r \geq 0$. Thus, we obtain

$$P_\epsilon(\tilde{\boldsymbol{x}}_*^r + \alpha \tilde{\boldsymbol{d}}^{r+1}) - P_\epsilon(\tilde{\boldsymbol{x}}_*^r) = \sum_{i=1}^{r} \left[p_\epsilon(\tilde{x}_i + \alpha \tilde{d}_i^{r+1}) - p_\epsilon(\tilde{x}_i) \right] + p_\epsilon(\alpha \tilde{d}^{r+1}) \geq 0$$

because (1) $p_\epsilon(\tilde{x}_i + \alpha \tilde{d}_i^{r+1}) - p_\epsilon(\tilde{x}_i) \geq 0$ for $i = 1,\ 2,\ \ldots,\ r$ (see Property 4 of Definition 3.1) and (2) $p_\epsilon(\alpha \tilde{d}^{r+1}) \geq 0$ for $\epsilon \geq \epsilon'$ (see Property 5 of Definition 3.1).

Using an approach similar to that in the proof of Theorem 3 of [47], we note that there are at most $(2n-r)$ edge directions of cone $\widetilde{\mathcal{F}}$ denoted as $\tilde{\boldsymbol{d}}^j$ with $j \in \{r+1,\ r+2,\ \ldots,\ 2n\}$. From the above analysis, we conclude that there exists a fixed $\alpha' > 0$ such that $P_\epsilon(\tilde{\boldsymbol{x}}_*^r + \alpha \tilde{\boldsymbol{d}}^j) \geq P_\epsilon(\tilde{\boldsymbol{x}}_*^r)$ for all $j \in \{r+1,\ r+2,\ \ldots,\ 2n\}$ and

for all $\alpha \in (0, \alpha']$. Now consider the set of feasible points $\widetilde{\mathcal{E}}$ given by

$$\widetilde{\mathcal{E}} = \left\{ \tilde{\boldsymbol{x}}_*^r, \quad \tilde{\boldsymbol{x}}_*^r + \alpha' \tilde{\boldsymbol{d}}^{r+1}, \quad \tilde{\boldsymbol{x}}_*^r + \alpha' \tilde{\boldsymbol{d}}^{r+2}, \quad \ldots, \quad \tilde{\boldsymbol{x}}_*^r + \alpha' \tilde{\boldsymbol{d}}^{2n} \right\}$$

The convex hull of set $\widetilde{\mathcal{E}}$, denoted by $H(\widetilde{\mathcal{E}})$, is the collection of all convex combinations of $\widetilde{\mathcal{E}}$, i.e., $\tilde{\boldsymbol{x}} \in H(\widetilde{\mathcal{E}})$ if and only if $\tilde{\boldsymbol{x}}$ can be expressed as

$$\tilde{\boldsymbol{x}} = \upsilon_r \tilde{\boldsymbol{x}}_*^r + \sum_{j=r+1}^{2n} \upsilon_j \left(\tilde{\boldsymbol{x}}_*^r + \alpha \tilde{\boldsymbol{d}}^j \right)$$

where $\sum_{j=r}^{2n} \upsilon_j = 1$ and $\upsilon_j \geq 0$ for $j = r, \ \ldots, \ 2n$ (see Definition 2.1.3 of [7]). Thus, for any $\tilde{\boldsymbol{x}} \in H(\widetilde{\mathcal{E}})$ and $\tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}}_*^r$, we must have $P_\epsilon(\tilde{\boldsymbol{x}}) \geq P_\epsilon(\tilde{\boldsymbol{x}}_*^r)$. Furthermore, one can always chose a sufficiently small but fixed $\alpha > 0$ such that

$$\left[B(\tilde{\boldsymbol{x}}_*^r, \alpha) \cap \widetilde{\mathcal{D}} \right] \subset H(\widetilde{\mathcal{E}})$$

Therefore, from Definition 3.3, $\tilde{\boldsymbol{x}}_*^r$ is a local minimizer and we conclude that the points in $\widetilde{\mathcal{S}}^r$ with $r \leq m$ are local minimizers of the problem in (3.11) □

We have identified local and global minimizers of the problem in (3.11) in terms of members of $\widetilde{\mathcal{S}}^r$ with $r \leq m$. It remains to address the case where $r > m$. As stated in the following lemma, members of $\widetilde{\mathcal{S}}^r$ with $r > m$ are saddle points.

Lemma 3.3 (Saddle Points of Problem in (3.11))**.** *The points in set $\widetilde{\mathcal{S}}^r$ with $m < r < 2n$ are* saddle *points of the problem in* (3.11).

Proof. Let $\widetilde{\boldsymbol{A}}^r$ denote the $m \times r$ matrix formed by collecting the corresponding r column vectors of $\widetilde{\boldsymbol{A}}$. Using a similar approach to that in Appendix A of [93], consider a point $\tilde{\boldsymbol{x}}^r = \tilde{\boldsymbol{x}}_*^r + \alpha \tilde{\boldsymbol{d}}_\varepsilon$ where $\alpha > 0$ and let $\tilde{\boldsymbol{d}}_\varepsilon$ with $||\tilde{\boldsymbol{d}}_\varepsilon||_2 = 1$ denote a direction vector given by

$$\tilde{\boldsymbol{d}}_\varepsilon = \tilde{\boldsymbol{d}} + \varepsilon \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T \tag{3.19}$$

where $\tilde{\boldsymbol{d}} \in \text{Nul}\left(\widetilde{\boldsymbol{A}}^r\right)$ is a direction in the null space of $\widetilde{\boldsymbol{A}}^r$. Suppose for now that $\varepsilon = 0$. Since matrix $\widetilde{\boldsymbol{A}}$ is of full row-rank on the basis of Proposition 3.1, $\dim\left[\text{Nul}\left(\widetilde{\boldsymbol{A}}^r\right)\right] = r - m$ [101], which implies that vector $\tilde{\boldsymbol{d}}$ is well defined for $r > m$ since a nontrivial null space exists for $\widetilde{\boldsymbol{A}}^r$. Hence, the Taylor series expansion of $P_\epsilon(\tilde{\boldsymbol{x}}^r)$ about the point

$\tilde{\boldsymbol{x}}_*^r$ is given by

$$P_\epsilon(\tilde{\boldsymbol{x}}^r) = P_\epsilon(\tilde{\boldsymbol{x}}_*^r) + \alpha[\boldsymbol{\nabla} P_\epsilon(\tilde{\boldsymbol{x}}_*^r)]^T \tilde{\boldsymbol{d}}_\varepsilon + \frac{\alpha^2}{2} \tilde{\boldsymbol{d}}_\varepsilon^T \boldsymbol{\nabla}^2 P_\epsilon(\tilde{\boldsymbol{x}}_*^r) \tilde{\boldsymbol{d}}_\varepsilon + O(\alpha^3) \tag{3.20}$$

The KKT condition in (3.14a) states that the result of the sum of gradient vector $\boldsymbol{\nabla} P_\epsilon(\tilde{\boldsymbol{x}}_*)$ and Lagrange multiplier vector $\boldsymbol{\mu}_*$ is in the column space of matrix $\widetilde{\boldsymbol{A}}^T$, i.e.,

$$[\boldsymbol{\nabla} P_\epsilon(\tilde{\boldsymbol{x}}_*) + \boldsymbol{\mu}_*] \in \mathrm{Col}(\widetilde{\boldsymbol{A}}^T) \tag{3.21}$$

where $\mathrm{Col}(\widetilde{\boldsymbol{A}}^T)$ denotes the column space of $\widetilde{\boldsymbol{A}}^T$. From (3.14e), we conclude that the entries of Lagrange multiplier vector $\boldsymbol{\mu}_*^r$ in (3.21) must be zero when corresponding entries of $\tilde{\boldsymbol{x}}_*^r$ are nonzero. Therefore, (3.21) can be simplified to

$$\boldsymbol{\nabla} P_\epsilon(\tilde{\boldsymbol{x}}_*^r) \in \mathrm{Col}(\widetilde{\boldsymbol{A}}^{rT})$$

and since subspaces $\mathrm{Col}(\widetilde{\boldsymbol{A}}^{rT})$ and $\mathrm{Nul}(\widetilde{\boldsymbol{A}}^r)$ are orthogonal, the inner product of a vector in the row space of $\widetilde{\boldsymbol{A}}^r$ with a vector in the null space of $\widetilde{\boldsymbol{A}}^r$ is zero [101]. Thus, the first term of the Taylor series expansion is zero and (3.20) can be simplified to

$$P_\epsilon(\tilde{\boldsymbol{x}}^r) = P_\epsilon(\tilde{\boldsymbol{x}}_*^r) + \frac{\alpha^2}{2} \tilde{\boldsymbol{d}}_\varepsilon^T \boldsymbol{\nabla}^2 P_\epsilon(\tilde{\boldsymbol{x}}_*^r) \tilde{\boldsymbol{d}}_\varepsilon + O(\alpha^3)$$

The second-term in the above Taylor series expansion is nonpositive as $p_\epsilon(\tilde{x}_i + \tilde{x}_{i+n})$ is a concave function from Property 3 in Definition 3.1. Therefore, $P_\epsilon(\tilde{\boldsymbol{x}}_*^r) \geq P_\epsilon(\tilde{\boldsymbol{x}}^r)$.

Consider now the Taylor series expansion of $P_\epsilon(\tilde{\boldsymbol{x}}^r)$ in the case where $\varepsilon \neq 0$ in (3.19), i.e., vector $\tilde{\boldsymbol{d}}_\varepsilon$ is given by direction $\tilde{\boldsymbol{d}}$ plus an arbitrary small perturbation $\varepsilon > 0$ in its kth entry. Under these circumstances, the Taylor series expansion is given by

$$P_\epsilon(\tilde{\boldsymbol{x}}^r) = P_\epsilon(\tilde{\boldsymbol{x}}_*^r) + \alpha[\boldsymbol{\nabla} P_\epsilon(\tilde{\boldsymbol{x}}_*^r)]^T \tilde{\boldsymbol{d}}_\varepsilon + O(\alpha^2)$$

By using the orthogonality of $\mathrm{Col}(\widetilde{\boldsymbol{A}}^{rT})$ and $\mathrm{Nul}(\widetilde{\boldsymbol{A}}^r)$, we have

$$P_\epsilon(\tilde{\boldsymbol{x}}^r) = P_\epsilon(\tilde{\boldsymbol{x}}_*^r) + \alpha\epsilon p'_\epsilon(\tilde{x}_k + \tilde{x}_{k\pm n}) + O(\alpha^2)$$

and since $p_\epsilon(\tilde{x}_k + \tilde{x}_{k\pm n})$ is nondecreasing and concave, $p'_\epsilon(\tilde{x}_k + \tilde{x}_{k\pm n})$ is nonincreasing and nonnegative. This implies that the second term in the Taylor expansion is nonnegative. Therefore, $P_\epsilon(\tilde{\boldsymbol{x}}_*^r) \leq P_\epsilon(\tilde{\boldsymbol{x}}^r)$.

For the neighborhood $B(\tilde{\boldsymbol{x}}_*^r, \alpha) \cap \widetilde{\mathcal{D}}$, one can choose sufficiently small but fixed

values of α and ε such that $\tilde{\boldsymbol{x}}^r$ belongs to $B(\tilde{\boldsymbol{x}}^r_*, \alpha) \cap \widetilde{\mathcal{D}}$. Furthermore, from the above analysis we conclude that $P_\epsilon(\tilde{\boldsymbol{x}}^r_*)$ is neither a maximum nor a minimum because there exist values assumed by $\tilde{\boldsymbol{x}}^r$ in the neighborhood that yield $P_\epsilon(\tilde{\boldsymbol{x}}^r_*) \leq P_\epsilon(\tilde{\boldsymbol{x}}^r)$ and $P_\epsilon(\tilde{\boldsymbol{x}}^r_*) \geq P_\epsilon(\tilde{\boldsymbol{x}}^r)$. Thus, $\tilde{\boldsymbol{x}}^r_*$ must be a saddle point. Therefore, points in $\widetilde{\mathcal{S}}^r$ with $m < r < 2n$ are saddle points of the problem in (3.11). □

In Lemmas 3.1 to 3.3, we have identified the stationary points of the problem in (3.11) as global and local minimizers and saddle points by looking at the number of nonzero coordinates of such points. Because the problem in (3.11) is closely related to a $\mathcal{P}$-class problem, we are able to relate their minimizers and, therefore, obtain results for our proposed class of recovery problems. This is done in terms of the following theorem.

Theorem 3.1 (Minimizers of $\mathcal{P}$-Class Problems)**.** *For a $\mathcal{P}$-class problem:*

1. *The stationary points with at most m nonzero coordinates are local minimizers;*

2. *A global minimizer is a stationary point with at most m nonzero coordinates;*

Proof. Let $\tilde{\boldsymbol{x}}^r_*$ denote a stationary point of a $\mathcal{P}$-class problem such as that in (3.7) that has r nonzero coordinates. Using a similar approach to that in Theorem 1 of [92], we can define a column vector $\boldsymbol{e}^*$ of length m given by

$$\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}^r_* - \boldsymbol{b} = \boldsymbol{e}_* \qquad \text{or} \qquad \widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}^r_* = \boldsymbol{e}_* + \boldsymbol{b} \tag{3.22}$$

If $\tilde{\boldsymbol{x}}^r_*$ is a stationary point of the problem in (3.7), then it also is a stationary point of the problem in (3.11) when column vector $\boldsymbol{c}$ as specified in (3.13) is given by $\boldsymbol{c} = \boldsymbol{e}_* + \boldsymbol{b}$. From Lemma 3.1, function $P_\epsilon(\tilde{\boldsymbol{x}})$ attains its minimum value when $\tilde{\boldsymbol{x}}$ is a stationary point with at most m nonzero coordinates. Thus, a global minimizer of the problem in (3.7) is a stationary point $\tilde{\boldsymbol{x}}^r_*$ with $r \leq m$. From Lemma 3.2, $P_\epsilon(\tilde{\boldsymbol{x}}^r_*)$ with $r \leq m$ is a local minimum of function $P_\epsilon(\tilde{\boldsymbol{x}})$. Thus, $\tilde{\boldsymbol{x}}^r_*$ must be a local minimizer of the problem in (3.7) when $r \leq m$. Furthermore, from Lemma 3.3, $P_\epsilon(\tilde{\boldsymbol{x}}^r_*)$ with $m < r < 2n$ is neither a maximum nor a minimum of function $P_\epsilon(\tilde{\boldsymbol{x}})$. Thus, $\tilde{\boldsymbol{x}}^r_*$ must be a saddle point of the problem in (3.7) when $m < r < 2n$.

From the above analysis, we conclude that for the problem in (3.7) (1) stationary points with at most m nonzero coordinates are local minimizers and (2) a global minimizer is a stationary point with at most m nonzero coordinates, which completes the proof. □

3.3 PLA Based Family of BP Methods

A family of SCF methods for the solution of $\mathcal{P}$-class problems will now be described. Consider a $\mathcal{P}$-class problem such as problem (BP_δ) in (1.7) with $p_\epsilon(|x_i|) \in \mathcal{P}$. It can be shown that $P_\epsilon(\boldsymbol{x})$ is a nonconvex function because the Hessian matrix of $P_\epsilon(\boldsymbol{x})$ is diagonal with entries that assume both positive and negative values. To render minimization of $P_\epsilon(\boldsymbol{x})$ tractable, we employ the approximation

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \sum_{i=1}^{n} \hat{p}_{\epsilon,x_i^{(k)}}(x_i) \tag{3.23}$$

where convex function $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ denotes an approximation of a $\mathcal{P}$-class function at $x_i = |x_i^{(k)}|$. As in Chapter 2, we will deduce an approximating function that possess the monotonic decreasing property (MDP) stated in Definition 2.1.

Using a similar approach to that in Sec. 2.2.2, we employ the first-order Taylor series approximation of a $\mathcal{P}$-class function at $x_i = |x_i^{(k)}|$ and $x_i = -|x_i^{(k)}|$

$$\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) = p_\epsilon(|x_i^{(k)}|) + \left(|x_i| - |x_i^{(k)}|\right) p'_\epsilon(|x_i^{(k)}|) \tag{3.24}$$

The convex function $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$ in (3.24) defines a PLA of $p_\epsilon(|x_i^{(k)}|)$ at $x_i = |x_i^{(k)}|$. In addition, such an approximating function is a nonsmooth function of x_i. Applicability of the PLA to the solution of $\mathcal{P}$-class problems is an immediate consequence of results proved in Sec. 2.2.2.

Corollary 3.1 (Monotonic Decreasing Piecewise-Linear Approximation)**.** *The convex approximating function* $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$ *in* (3.24) *has the MDP at* $x_i = |x_i^{(k)}|$.

Proof. As in the proof of Proposition 2.2, we note that $p_\epsilon(|x_i|) \in \mathcal{P}$ is a concave function for $x_i \in (0, \infty)$ (see Property 3 in Definition 3.1) and, therefore, the conditions in 2.8a and (2.34) hold true. Thus, $\hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i)$ has the MDP at $x_i = |x_i^{(k)}|$. □

Corollary 3.2 (Best Convex Approximation)**.** *Let* $\mathcal{A}$ *denote the class of all convex approximating functions with the MDP at* $x_i = |x_i^{(k)}|$*, and let* $\hat{p}_{\epsilon,x_i^{(k)}}(x_i) \in \mathcal{A}$ *denote a convex approximating function in class* $\mathcal{A}$*. The condition*

$$\hat{p}_{\epsilon,x_i^{(k)}}(x_i) \geq \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) \geq p_\epsilon(|x_i|), \qquad \forall x_i \in \mathbb{R} \wedge x_i \neq x_i^{(k)} \tag{3.25}$$

holds true for the PLA of a $\mathcal{P}$*-class function .*

Proof. As in the proof of Proposition 2.3, it suffices to show that the condition in (2.37) holds true. The conditions in (2.38) and (2.39) hold true for function $\hat{p}_{\epsilon,x_i^{(k)}}(x_i)$ and, therefore, (2.41) is applicable to the cases where $x_i \in (0,\infty)$ and $x_i \in (-\infty,0)$. Thus, the condition in (2.37) holds true, which completes the proof. □

The results presented in Corollaries 3.1 and 3.2 show that (1) the PLA is applicable to SCF methods which can be used for the solution of $\mathcal{P}$-class problems, and (2) the PLA is the best convex approximation of a $\mathcal{P}$-class function as it provides the least upper bound on such function.

We are now in a position to describe the proposed family of solution methods. Using the notation of Sec. 1.2.3, a $\mathcal{P}$-class problem such as problem (BP_δ) in (1.7) with $p_\epsilon(|x_i|) \in \mathcal{P}$ entails minimizing nonconvex function $F(\boldsymbol{x}) = P_\epsilon(\boldsymbol{x})$ over convex set $\mathcal{X} = \mathcal{K}$, where $\mathcal{K}$ is the closed ball in $\mathbb{R}^n$ under an affine mapping given by

$$\mathcal{K} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \delta\}$$

We can find a solution of a $\mathcal{P}$-class problem by letting $\boldsymbol{x}^{(0)} \in \mathcal{K}$ and then applying the update formula in (1.22) with $\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ where $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is based on the PLA in (3.24). Hence, the update formula in (1.22) can be written as

$$\boldsymbol{x}^{(k+1)} = \arg \underset{\boldsymbol{x}\in\mathcal{K}}{\text{minimize}}\ \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{3.26}$$

where

$$\begin{aligned}\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) &= \sum_{i=1}^{n} \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) \\ &= \sum_{i=1}^{n} \left[p_\epsilon(|x_i^{(k)}|) + \left(|x_i| - |x_i^{(k)}|\right) p'_\epsilon(|x_i^{(k)}|)\right]\end{aligned}$$

A signal recovery method in which a solution of a $\mathcal{P}$-class problem is obtained by computing the update formula in (3.26) will hereafter be called a $\mathcal{P}$-class method. Specifically, if we let $p_\epsilon(|x_i|)$ assume the form in (1.4), (1.5), or (1.6), then the $\mathcal{P}$-class method will be denoted as the $\mathcal{P}$-$\mathrm{CM}_{\ell_p^p}$, $\mathcal{P}$-$\mathrm{CM}_{\ln}$, or $\mathcal{P}$-$\mathrm{CM}_{\mathrm{SCAD}}$, respectively.

3.3.1 Proposed FOS

Below we reformulate the convex subproblems in (3.26) such that the update formula can be efficiently computed.

In the minimization of $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}$, terms that do not involve $\boldsymbol{x}$ are constant and can be dropped as they do not change the solution. Therefore, each subproblem in (3.26) can be written as

$$\underset{\boldsymbol{x}\in\mathcal{K}}{\text{minimize}}\ \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{3.27}$$

where

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \sum_{i\in\mathcal{I}_n} p'_\epsilon(|x_i^{(k)}|)|x_i|$$

From Properties 3 and 4 in Definition 3.1, $\mathcal{P}$-class functions are nondecreasing and concave. Therefore, $p'_\epsilon(|x_i|)$ is nonincreasing and nonnegative and

$$p'_\epsilon(|x_i^{(k)}|)\,|x_i| = |p'_\epsilon(|x_i^{(k)}|)\,x_i|$$

Thus, function $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ can be written as the weighted ℓ_1-norm of vector $\boldsymbol{x}$

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \left\|\boldsymbol{W}^{(k)}\boldsymbol{x}\right\|_1 \tag{3.28}$$

where $\boldsymbol{W}^{(k)} = \text{diag}(w_1^{(k)},\ w_2^{(k)},\ \ldots,\ w_n^{(k)})$ is a diagonal weight matrix and $w_i^{(k)} = p'_\epsilon(|x_i^{(k)}|)$ denotes a nonnegative weight.

Because the problem in (3.27) is equivalent to a weighted ℓ_1-norm minimization problem, we can employ recent state-of-the-art FOSs such as NESTA [9] for its solution. In other words, the nonsmooth function $\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ in (3.28) has the conjugate (or dual) representation specified in (1.31) and, therefore, it can be approximated by (1.32). Furthermore, there exists an analytical solution for the maximization problem defined by such an approximation (see Sec. 4.3 of [61]) and we can, therefore, obtain an approximation of the function in (3.28) given by

$$\widehat{P}_{\mu,\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) = \underset{\boldsymbol{u}\in\mathcal{Q}}{\text{maximize}}\ \left(\langle\boldsymbol{u},\boldsymbol{W}\boldsymbol{x}\rangle - \frac{\mu}{2}||\boldsymbol{u}||_2^2\right) = \sum_{i=1}^{n} h_\mu(w_i^{(k)}x_i) \tag{3.29}$$

where $\mu > 0$ is a smoothness parameter and $h_\mu(x)$ denotes the Huber penalty function

given by [61]

$$h_\mu(x) = \begin{cases} x^2/2\mu, & |x| \leq \mu \\ |x| - \mu/2, & |x| > \mu \end{cases}$$

As a result of (3.29), the smooth convex optimization problem given by

$$\underset{\boldsymbol{x}\in\mathcal{K}}{\text{minimize}}\ \widehat{P}_{\mu,\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \tag{3.30}$$

can be used to find a solution of the nonsmooth problem in (3.27) when μ is appropriately chosen. In addition, the function in (3.29) is Lipschitz continuous and, therefore, the problem in (3.30) can be solved efficiently by using an optimal first-order method with convergence rate given by (1.21) (see Secs. 2.2 and 6.1 of [9] for details). The proposed solver uses matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ in matrix-vector operations and, therefore, can handle large-scale weighted ℓ_1-norm minimization problems because (1) there is no need for storage of such matrices, and (2) matrix-vector operations can be carried out with fast algorithms in the case of orthogonal ensembles.

In summary, the proposed FOS can be used to find the solution of nonsmooth convex problems such as the one in (3.27). Thus, the update formula in (3.26) can be efficiently computed. Furthermore, if sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (3.26) converges to a solution, large-scale $\mathcal{P}$-class problems can be solved. A convergence analysis is presented in the following subsection.

3.3.2 Convergence analysis

From Corollary 3.1, the function in (3.24) has the MDP at $x_i = |x_i^{(k)}|$ and, therefore, the conditions

$$\sum_{i=1}^{n} \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i) \geq \sum_{i=1}^{n} p_\epsilon(|x_i|), \qquad \forall x_i \in \mathbb{R} \tag{3.31}$$

and

$$\sum_{i=1}^{n} \hat{p}^{\,pl}_{\epsilon,x_i^{(k)}}(x_i^{(k)}) = \sum_{i=1}^{n} p_\epsilon(|x_i^{(k)}|) \tag{3.32}$$

hold true (see Definition 2.1). Combining (1.2), (3.23), (3.31), and (3.32), we obtain

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}) \geq P_\epsilon(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathbb{R}^n \wedge \boldsymbol{x} \neq \boldsymbol{x}^{(k)} \tag{3.33}$$

and

$$\widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k)}) = P_{\epsilon}(\boldsymbol{x}^{(k)}) \tag{3.34}$$

As a result of (3.33) and (3.34), the inequalities

$$P_{\epsilon}(\boldsymbol{x}^{(k+1)}) \leq \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k+1)}) < \widehat{P}_{\epsilon,\boldsymbol{x}^{(k)}}(\boldsymbol{x}^{(k)}) \leq P_{\epsilon}(\boldsymbol{x}^{(k)}) \tag{3.35}$$

can be shown to hold true by applying the update formula in (3.26) and sequence $\left\{P_{\epsilon}(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ is deemed a monotonically decreasing sequence. In addition, from Property 5 in Definition 3.1, function $P_{\epsilon}(\boldsymbol{x})$ is bounded from below by a real number proportional to A and, as a result, $\left\{P_{\epsilon}(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ is bounded. Therefore, $\left\{P_{\epsilon}(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ converges because every bounded monotonic decreasing sequence converges (see monotonic sequence theorem, p. 710 of [100]).

Let $\mathcal{U}$ denote an infinite set of nonnegative integers in their natural order, and let $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathcal{U}}$ denote a subsequence of $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (3.26). For instance, if we let

$$\mathcal{U} = \{1,\ 3,\ 7,\ 10,\ 13,\ \ldots\}$$

then

$$\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathcal{U}} = \left\{\boldsymbol{x}^{(1)},\ \boldsymbol{x}^{(3)},\ \boldsymbol{x}^{(7)},\ \boldsymbol{x}^{(10)},\ \boldsymbol{x}^{(13)},\ \ldots\right\}$$

(see Sec. 4.2 of [115]). Limit points of convergent subsequences $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathcal{U}}$ can be shown to satisfy the optimality conditions presented in Sec. 3.2.2 by relating the proposed family of solution methods to the class of majorization-minimization (MM) methods.

In an MM method, the solution of an optimization problem is obtained by solving several related subproblems in sequence and, therefore, such an approach entails an update formula similar to that in (1.22) (see [62, 70, 71] for details). In the context of MM methods, function $\widehat{F}_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ in (1.22) is a called a *majorizing* function. It can easily be shown that SCF and MM methods are closely related because a convex approximation function with the MDP property, as stated in Definition 2.1, is a majorizing function. The converse, however, is not true. Thus, SCF methods based on convex approximation functions with the MDP property belong to the class of MM methods. The numerical stability of MM methods is confirmed by strong theoretical results presented by Jacobson *et al.* in [64]. These results include convergence conditions based on upper curvature bounds that are often more easily verifiable than previously proposed convergence conditions.

By using the convergence conditions (R1), (R2), (R3), and (C6) of [64] in conjunction with the optimality conditions presented in Sec. 3.2.2, we can now show that limit points of convergent subsequences $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathcal{U}}$ are solutions of a $\mathcal{P}$-class problem. This convergence result is presented in terms of the following theorem.

Theorem 3.2 (Convergence to a Minimizer)**.** *The limit point of any convergent subsequence* $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathcal{U}}$ *is a local minimizer of a* $\mathcal{P}$*-class problem such as problem* (BP_δ) *in* (1.7) *with* $p_\epsilon(|x_i|)\in\mathcal{P}$*. In addition, such a local minimizer is a point with at most* m *nonzero coordinates.*

Proof. Consider the equivalent smooth formulation of problem (BP_δ) in (1.7) with $p_\epsilon(|x_i|)\in\mathcal{P}$ as given in (3.7). By using the variable transformation in (3.5), it can easily be shown that the update formula in (3.26) can be written in terms of the equivalent update formula

$$\tilde{\boldsymbol{x}}^{(k+1)} = \arg\ \underset{\tilde{\boldsymbol{x}}\in\widetilde{\mathcal{K}}}{\text{minimize}}\ \widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}}) \tag{3.36}$$

where

$$\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}^{(k)}}(\tilde{\boldsymbol{x}})=\sum_{i=1}^{n}\left\{p_\epsilon(\tilde{x}_i^{(k)}+\tilde{x}_{i+n}^{(k)})+\left[\tilde{x}_i+\tilde{x}_{i+n}-(\tilde{x}_i^{(k)}+\tilde{x}_{i+n}^{(k)})\right]p'_\epsilon(\tilde{x}_i^{(k)}+\tilde{x}_{i+n}^{(k)})\right\} \tag{3.37}$$

(see Sec. 3.2.1 for details). Let $\left\{\tilde{\boldsymbol{x}}^{(k)}\right\}_{k\in\mathcal{U}}$ denote a subsequence of $\left\{\tilde{\boldsymbol{x}}^{(k)}\right\}_{k\in\mathbb{N}}$ in (3.36). The convergence of $\left\{\tilde{\boldsymbol{x}}^{(k)}\right\}_{k\in\mathcal{U}}$ to a stationary point $\tilde{\boldsymbol{x}}_*$ of the problem in (3.7) can be established as follows. First, we note that set $\widetilde{\mathcal{K}}$ in (3.9) defines a *closed convex set*. Set $\widetilde{\mathcal{M}}$ in (3.10) can be written as the intersection of $2n$ sets of the form

$$\widetilde{\mathcal{M}} = \bigcap_{k\in\widetilde{\mathcal{I}}_{2n}} \widetilde{\mathcal{M}}_k$$

where $\widetilde{\mathcal{M}}_k = \left\{\tilde{\boldsymbol{x}}\in\mathbb{R}^{2n} : \boldsymbol{c}_k^T\tilde{\boldsymbol{x}}\leq 0\right\}$ is a half-space (see Sec. 2.2.1 of [16]) and $\boldsymbol{c}_k$ is a $2n$-length column vector with entries $c_{k,i}$ given by

$$c_{k,i} = \begin{cases} -1, & i=k \\ 0, & \text{otherwise} \end{cases}$$

Since a half-space defines a closed and convex set [16] and the intersection of closed sets defines a closed set, set $\widetilde{\mathcal{M}}$ in (3.10) is a *closed convex set*. Thus, the feasible

set $\widetilde{\mathcal{C}}$ in (3.8) is *closed*. Second, we note that for the majorizing function defined by (3.37), the relation $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}}) = P_\epsilon(\tilde{\boldsymbol{x}})$ holds true. This implies that the gradient of the majorizing function is equal to the gradient of the objective function, namely, $\boldsymbol{\nabla}\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}}) = \boldsymbol{\nabla}P_\epsilon(\tilde{\boldsymbol{x}})$. Third, we note that the Hessian matrix of $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}})$ is defined by a block matrix given by

$$\boldsymbol{\nabla}^2\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}}) = \boldsymbol{H} = \begin{bmatrix} \text{diag}(\boldsymbol{h}) & \text{diag}(\boldsymbol{h}) \\ \text{diag}(\boldsymbol{h}) & \text{diag}(\boldsymbol{h}) \end{bmatrix}$$

where $\boldsymbol{h} \in \mathbb{R}^n$ with $h_i = p''_\epsilon(\tilde{x}_i + \tilde{x}_{i+n})$ for $i \in \mathcal{I}_n$. Thus, we obtain a closed-form expression for the Frobenius norm of matrix $\boldsymbol{H}$ given by

$$||\boldsymbol{H}||_F = \sqrt{4\sum_{i\in\mathcal{I}_n} |p''_\epsilon(\tilde{x}_i + \tilde{x}_{i+n})|^2}$$

From Property 2 in Definition 3.1, we have $|p''_\epsilon(\tilde{x}_i + \tilde{x}_{i+n})| \leq \rho$ for $i \in \mathcal{I}_n$ where $\rho \in \mathbb{R}_0^+$. Therefore, we obtain

$$\|\boldsymbol{H}\|_F \leq \sqrt{4\sum_{i\in\mathcal{I}_n} \rho^2} \tag{3.38}$$

The relation in (3.38) implies that the Hessian matrix of function $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}})$ is uniformly bounded (see assumption AM.4, p. 122 of [34]).

In summary, (1) the feasible set $\widetilde{\mathcal{C}}$ of the problem in (3.7) is *closed*, (2) the gradient of the majorizing function $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}})$ in (3.37) is equal to the gradient of the objective function $P_\epsilon(\tilde{\boldsymbol{x}})$ of the problem in (3.7), and (3) the curvature of each majorizing function $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}})$ in (3.37) is uniformly upper bounded, which is equivalent to saying that the Frobenius norm of the Hessian matrix of function $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}})$ has a bound such as that in (3.38). Such curvature bounds of $\widehat{P}_{\epsilon,\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}})$ hold true because condition (C6) of [64] is equivalent to assumption AM.4 of [34], see discussion on the relation between MM and trust-region methods at beginning of Sec. IV of [64]. Therefore, conditions (R1), (R2), (R3), and (C6) of [64] for the convergence of an MM method applied to smooth and constrained nonconvex optimization problems also hold true for the proposed family of methods applied to smooth $\mathcal{P}$-class problems. From Theorem 4.1 of [64], the limit point of any convergent subsequence $\{\tilde{\boldsymbol{x}}^{(k)}\}_{k\in\mathcal{U}}$ is a *stationary point* $\tilde{\boldsymbol{x}}_*$ of the problem in (3.7).

If $\tilde{\boldsymbol{x}}_*^r$ is a stationary point of the problem in (3.7), then it is also a stationary point of the problem in (3.11) when $\boldsymbol{c} = \boldsymbol{e}_* + \boldsymbol{b}$ where $\boldsymbol{e}^*$ is given by (3.22). Consider the case where $\tilde{\boldsymbol{x}}_*^r$ is a saddle point, i.e., when $m < r < 2n$ on the basis of Lemma 3.3. Stationary point $\tilde{\boldsymbol{x}}_*^r$ must satisfy the KKT conditions in (3.14) and it must belong to the solution set $\widetilde{\mathcal{H}}$ in (3.13). Furthermore, because $\tilde{\boldsymbol{x}}_*^r$ is a solution of the linear system of equations, it can be expressed as $\tilde{\boldsymbol{x}}_*^r = \tilde{\boldsymbol{x}}_p + \tilde{\boldsymbol{x}}_{\text{Nul}(\widetilde{\boldsymbol{A}})}$ where $\tilde{\boldsymbol{x}}_p$ is a particular solution in $\widetilde{\mathcal{H}}$ and $\tilde{\boldsymbol{x}}_{\text{Nul}(\widetilde{\boldsymbol{A}})}$ is in the null space of matrix $\widetilde{\boldsymbol{A}}$ [101]. Geometrically, set $\widetilde{\mathcal{H}}$ can be viewed as a linear variety, i.e., the set defined by the translation of a subspace [73], since each element of the linear variety $\widetilde{\mathcal{H}}$ defines a translation of the null space of $\widetilde{\boldsymbol{A}}$ by the particular solution $\tilde{\boldsymbol{x}}_p$. Now, by using a similar argument to that in Appendix A of [93], we note that it is simultaneously required from (3.14) that the saddle point $\tilde{\boldsymbol{x}}_*^r$ lie in a linear variety of dimension $(r-m) = \dim[\text{Nul}(\widetilde{\boldsymbol{A}}^r)]$ and also that the gradient vector $\boldsymbol{\nabla} P_\epsilon(\tilde{\boldsymbol{x}}_*^r)$ lie in $\text{Col}(\widetilde{\boldsymbol{A}}^{rT})$ which is a subspace of dimension m. It so happens that generic members of the solution set of a linear system of equations defined by a full row-rank matrix, such as $\widetilde{\mathcal{H}}$ in (3.13), do not satisfy these conditions [93]. Furthermore, the probability of a subsequence of feasible points of this solution set, such as $\left\{\tilde{\boldsymbol{x}}^{(k)}\right\}_{k\in\mathcal{U}}$, converging to $\tilde{\boldsymbol{x}}_*^r$ is zero when $m < r < 2n$ (see proof of Corollary 3 in [51]).

It should be mentioned that even if convergence to a saddle point were theoretically possible, in practice the subsequence $\left\{\tilde{\boldsymbol{x}}^{(k)}\right\}_{k\in\mathcal{U}}$ would be highly unlikely to converge to the exact value of a saddle point, $\tilde{\boldsymbol{x}}_*$, due to roundoff errors and, consequently, if a point in the neighborhood of a saddle point were to be reached, a descent direction would exist which would cause any descent algorithm to locate another stationary point. In practice, a descent algorithm will stop only when a local minimum is located whether the problem has a saddle point or not.

Based on the above arguments, the limit of any convergent subsequence $\left\{\tilde{\boldsymbol{x}}^{(k)}\right\}_{k\in\mathcal{U}}$ is a stationary point $\tilde{\boldsymbol{x}}_*^r$ with $r \leq m$. From Theorem 3.1, such a stationary point is a local minimizer of the problem in (3.7) with at most m nonzero coordinates. Furthermore, because of the equivalence between the update formulas in (3.26) and (3.36), we conclude that the same convergence result also holds true for a $\mathcal{P}$-class problem such as problem (BP_δ) in (1.7) with $p_\epsilon(|x_i|) \in \mathcal{P}$, which completes the proof. □

3.4 Simulation Results

We now present simulation results to evaluate the proposed and corresponding competing methods in terms of their capability of recovering signals in a wide range of test problems. The signal-recovery methods were evaluated following the experimental protocol discussed in Sec. 1.3. The average ℓ_∞ reconstruction error, probability of perfect recovery (PPR), and minimum required fraction (MRF), m/s, for perfect recovery were employed as reconstruction performance (RP) metrics. The difference between the Euclidean distance $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ and the estimate of the square root of the measurement noise energy δ was employed as the measurement consistency (MC) metric. The average CPU time in seconds was employed as the computational cost (CC) metric. These metrics were estimated by carrying out the recovery process 100 times using Gaussian or orthogonal measurement ensembles.

Each measurement vector $\boldsymbol{b}$ was generated by applying a renormalized Gaussian or discrete Fourier transform (DFT) matrix to the signal of interest $\boldsymbol{x}^0$. The length of vector $\boldsymbol{b}$ was assumed to be $m = n/8$ and the sparsity of $\boldsymbol{x}^0$ was assumed to be in the range $\frac{m}{100} \leq s \leq \frac{m}{2}$. The MRF, m/s, for perfect recovery was estimated by finding the minimum value of s in that range where PPR = 1. In the recovery of noisy signals, the s nonzero values of $\boldsymbol{x}^0$ were generated as in (1.37) with parameter $\kappa = 1$. This results in signals with a dynamic range (DR) of 20 dB where the absolute values of its nonzero entries are distributed between 1 and 10. Each measurement vector $\boldsymbol{b}$ was obtained using a Gaussian vector $\boldsymbol{z}$ with $\sigma_{\boldsymbol{z}} = 1 \times 10^{-4}$ and perfect signal recovery was declared when $\nu = 5 \times 10^{-2}$ in (1.36). In the recovery of noiseless signals, the s nonzero values of $\boldsymbol{x}^0$ were chosen randomly from a zero-mean Gaussian distribution of unit variance and perfect signal recovery was declared when $\nu = 1 \times 10^{-3}$ in (1.36).

The proposed update formula in (3.26) was applied with the zero vector as initial point $\boldsymbol{x}^{(0)} \in \mathcal{K}$ and the stopping criterion

$$\left\| P_\epsilon(\boldsymbol{x}^{(k+1)}) - P_\epsilon(\boldsymbol{x}^{(k)}) \right\|_2 \leq \varepsilon_c$$

was used for the computation of $\boldsymbol{x}^{(k+1)}$ where $\varepsilon_c = 1 \times 10^{-3}$. In the proposed FOS, the approximation in (3.29) was obtained by using $\mu = 1 \times 10^{-6}$ and $\mu = 1 \times 10^{-4}$ in the cases of noiseless and noisy signals, respectively. Computing the update formula in (3.26) requires the prior selection of several parameters such as the regularization parameter ϵ in (1.4) to (1.6), parameter p for the weighted ϵ-ℓ_p^p-norm of $\boldsymbol{x}$ in (1.4) and

parameter α for the smoothly-clipped absolute deviation (SCAD) function in (1.6). Empirical evidence suggests that the value of ϵ should be approximately of the same order of magnitude as the absolute values of the coordinates of $\boldsymbol{x}^0$ [23,24,26,104,105]. In the case of noiseless signals where nonzero values of $\boldsymbol{x}^0$ were chosen randomly from a zero-mean Gaussian distribution of unit variance, we used $\epsilon = 1$. In the case of noisy signals where nonzero values of $\boldsymbol{x}^0$ were generated as in (1.37) with a resulting DR of 20 dB, we used $\epsilon = 10$. Empirical evidence suggests that decreasing the value of p in (1.4) below 1 down to 0.5 provides continuously improving RP metrics [24,26,37]. We observed the same effect for our method and, for this reason, we used $p = 0.5$. Statistical analysis conducted in [42] demonstrates that the use of $\alpha = 3.7$ in (1.6) is a near-optimal value for various variable selection problems in statistics. Simulation results suggest that this result also applies to CS reconstruction problems [104,105]. Thus, we used $\alpha = 3.7$.

All experiments were run on a Dell Precision 670 workstation with two 3.2 GHz dual-core Intel Xeon processors and 4 Gb of RAM using the 64-bit Linux MATLAB Version 7.13 (R2011b). Software that is publicly available online was used for the competing methods.[1] We used the values suggested in [11,18,37,46] for the several parameters of the software obtained with the exception of parameters also used in the proposed $\mathcal{P}$-class methods such as the regularization parameter ϵ in (1.4) to (1.6), the value of p in (1.4), and the value of α in (1.6). In effect, the same values of ϵ, p, and α were used for the proposed and competing methods for comparison purposes. For the continuation procedure used in the DC-family of methods, a decreasing sequence $\{\lambda_1, \ldots, \lambda_5\}$ with $\lambda_1 = \left\|\boldsymbol{A}^T\boldsymbol{b}\right\|_{\ell_\infty}$ and $\lambda_5 = 1 \times 10^{-4} \left\|\boldsymbol{A}^T\boldsymbol{b}\right\|_{\ell_\infty}$ was applied, which turned out to yield the best results in simulations.

[1] The codes for the competing methods were obtained from the respective author's Web pages:

Regularized least-squares (RLS) methods: Difference-of-two-convex-functions (DC)-family from A. Rakotomamonjy at http://asi.insa-rouen.fr/enseignants/~arakotom/.

Least absolute shrinkage and selection operator (LASSO) methods: Spectral projected-gradient ℓ_1-norm (SPGL1) from M. P. Friedlander at http://www.cs.ubc.ca/~mpf/spgl1/.

Basis pursuit (BP) methods: ℓ_1-Magic from J. Romberg at http://users.ece.gatech.edu/~justin/l1magic/, and iteratively reweighted least squares (IRWLS) from M. Fornasier at http://www.ricam.oeaw.ac.at/people/page/fornasier/.

3.4.1 Evaluation of proposed family of BP methods

We first carried out recovery simulations using orthogonal ensembles to evaluate the relative merits of the proposed $\mathcal{P}$-class methods in the case of noisy signals.

The RP of $\mathcal{P}$-class methods for signals of size $n = 65,536$ is compared in Fig. 3.1. As can be seen, the $\mathcal{P}$-CM$_{\ln}$ method achieved superior RP relative to that of the $\mathcal{P}$-CM$_{\ell_p^p}$ and $\mathcal{P}$-CM$_{\text{SCAD}}$ methods. For instance, the average ℓ_∞ reconstruction error of the $\mathcal{P}$-CM$_{\ln}$, $\mathcal{P}$-CM$_{\ell_p^p}$, and $\mathcal{P}$-CM$_{\text{SCAD}}$ methods in Fig. 3.1a were 0.0033, 0.5072, and 0.0244, respectively, for $s \leq 2,542$. The MRF for perfect reconstruction in Fig. 3.1b

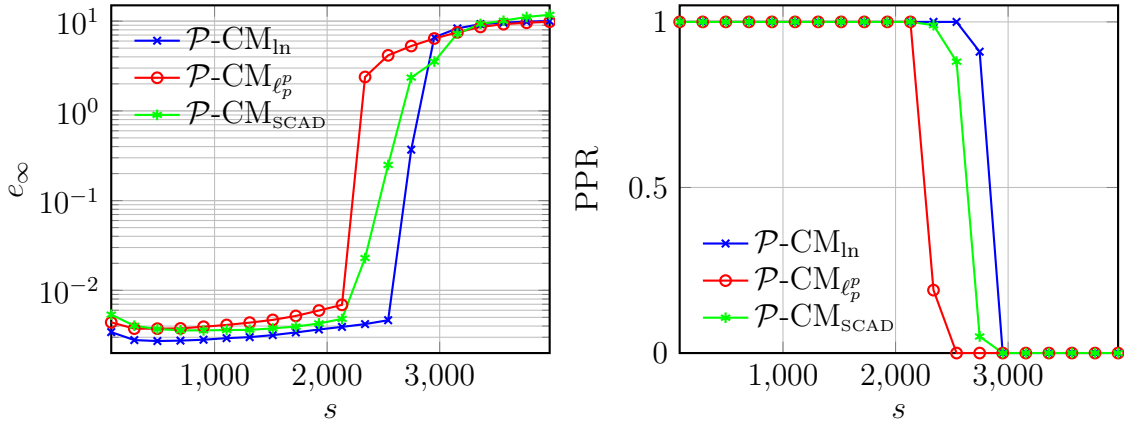


Figure 3.1: RP metrics of $\mathcal{P}$-class methods: (a) ℓ_∞ recovery error and (b) PPR.

has dropped from $8,192/2,132 \approx 3.8$ in the $\mathcal{P}$-CM$_{\text{SCAD}}$ and $\mathcal{P}$-CM$_{\ell_p^p}$ methods to $8,192/2,542 \approx 3.2$ in the $\mathcal{P}$-CM$_{\ln}$ method.

We compared the average number of points k computed for sequence $\left\{P_\epsilon(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ to converge. The results obtained are plotted in Fig. 3.2. As can be seen, convergence is usually attained very fast since roughly four points are computed, on average, in simulations where the sparse signal is always perfectly reconstructed. The average number of points k computed increases for increasing values of s for simulations where the signal is not always perfectly reconstructed. In such simulations, the $\mathcal{P}$-CM$_{\text{SCAD}}$ method is the fastest. For instance, the average number of points computed by the $\mathcal{P}$-CM$_{\ln}$, $\mathcal{P}$-CM$_{\ell_p^p}$, and $\mathcal{P}$-CM$_{\text{SCAD}}$ methods were found to be approximately 99.5, 81.7, and 70.5, respectively, for $s = 3,157$.

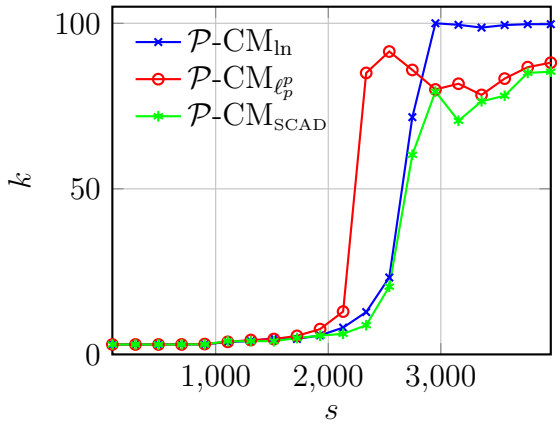


Figure 3.2: Convergence rate of $\mathcal{P}$-class methods.

3.4.2 Comparison of the proposed family of BP methods with state-of-the-art competing methods

We carried out recovery simulations using Gaussian and orthogonal ensembles to evaluate the proposed and competing methods for the recovery of noiseless and noisy signals. The results obtained for noiseless signals of length $n = 2,048$ and $n = 4,096$ with sparsity $s \leq 45$ and $s \leq 96$, respectively, are summarized in Table 3.1. As can be seen, the $\mathcal{P}$-CM$_{\ell_p^p}$ method achieved superior RP, reduced CC, and increased MC relative to those of the ℓ_1-Magic and IRWLS methods for the case of noiseless signals and Gaussian ensembles.

	RP		CC	MC		
n	e_∞	MRF	CPU time	median of $\|\|\boldsymbol{Ax}^*-\boldsymbol{b}\|\|_2$	δ	method
	0.0035	7.7	5.3	1.85×10^{-8}		ℓ_1-Magic
2,048	0.0470	7.7	6.1	3.8×10^{-15}	0	IRWLS
	2.7×10^{-5}	5.7	2.0	3.73×10^{-15}		$\mathcal{P}$-CM$_{\ell_p^p}$
	0.0089	6.2	24.9	9.69×10^{-8}		ℓ_1-Magic
4,096	0.0893	7.3	31.7	3.43×10^{-14}	0	IRWLS
	3.2×10^{-5}	5.3	9.0	7.53×10^{-15}		$\mathcal{P}$-CM$_{\ell_p^p}$

Table 3.1: Summary of results for noiseless signals and Gaussian ensembles.

The results obtained for noisy signals of length $n = 16,384$ and $n = 32,768$ with

sparsity $s \leq 632$ and $s \leq 1,265$, respectively, are summarized in Table 3.2. As can be seen, the $\mathcal{P}$-CM$_{\text{ln}}$ method achieved superior RP and increased MC relative to those of the ℓ_1-Magic and IRWLS methods for the case of noisy signals and orthogonal ensembles. The CC of the $\mathcal{P}$-CM$_{\text{ln}}$ method was slightly increased relative to that of the SPGL1 method and considerably reduced relative to that of the DC$_{\text{ln}}$ method.

	RP		CC	MC		
n	e_∞	MRF	CPU time	median of $\|\|\boldsymbol{Ax}^*-\boldsymbol{b}\|\|_2$	δ	method
	1.69	9.1	7.1	46×10^{-4}		SPGL1
16,384	1.53	4.8	195.5	74×10^{-4}	45×10^{-4}	DC$_{\text{ln}}$
	0.003	3.2	10.2	45×10^{-4}		$\mathcal{P}$-CM$_{\text{ln}}$
	1.77	7.4	13.7	5×10^{-4}		SPGL1
32,768	1.53	4.8	318.1	112×10^{-4}	64×10^{-4}	DC$_{\text{ln}}$
	0.0032	3.2	19.2	64×10^{-4}		$\mathcal{P}$-CM$_{\text{ln}}$

Table 3.2: Summary of results for noisy signals and orthogonal ensembles.

The results summarized in Tables 3.1 and 3.2 are described in detail in the following subsections.

Results for noiseless signals and Gaussian ensembles

RP and CC of the $\mathcal{P}$-CM$_{\ell_p^p}$ method are compared with those of the IRWLS and ℓ_1-Magic methods in Fig. 3.3. As can be seen in Figs. 3.3a to 3.3d, the proposed method achieved superior RP relative to that of the competing methods for different signal sizes. For instance, when $n = 2,048$ the average ℓ_∞ reconstruction error of the $\mathcal{P}$-CM$_{\ell_p^p}$, IRWLS, and ℓ_1-Magic methods in Fig. 3.3a were 2.68×10^{-5}, 0.0470, and 0.0035, respectively, for $s \leq 45$ while the MRF for perfect reconstruction in Fig. 3.3c has dropped from $256/33 \approx 7.7$ in the IRWLS and ℓ_1-Magic methods to $256/45 \approx 5.7$ in the $\mathcal{P}$-CM$_{\ell_p^p}$ method. On the other hand, when $n = 4,096$ the average ℓ_∞ reconstruction error of the $\mathcal{P}$-CM$_{\ell_p^p}$, IRWLS, and ℓ_1-Magic methods in Fig. 3.3b were 3.2×10^{-5}, 0.0893, 0.0089, respectively, for $s \leq 96$ while the MRF for perfect reconstruction in Fig. 3.3d has dropped from $512/70 \approx 7.3$ and $512/83 \approx 6.8$ in IRWLS and ℓ_1-Magic methods, respectively, to $512/96 \approx 5.3$ in the $\mathcal{P}$-CM$_{\ell_p^p}$ method. As can be seen in Figs. 3.3e and 3.3f, the superior RP of the proposed method comes

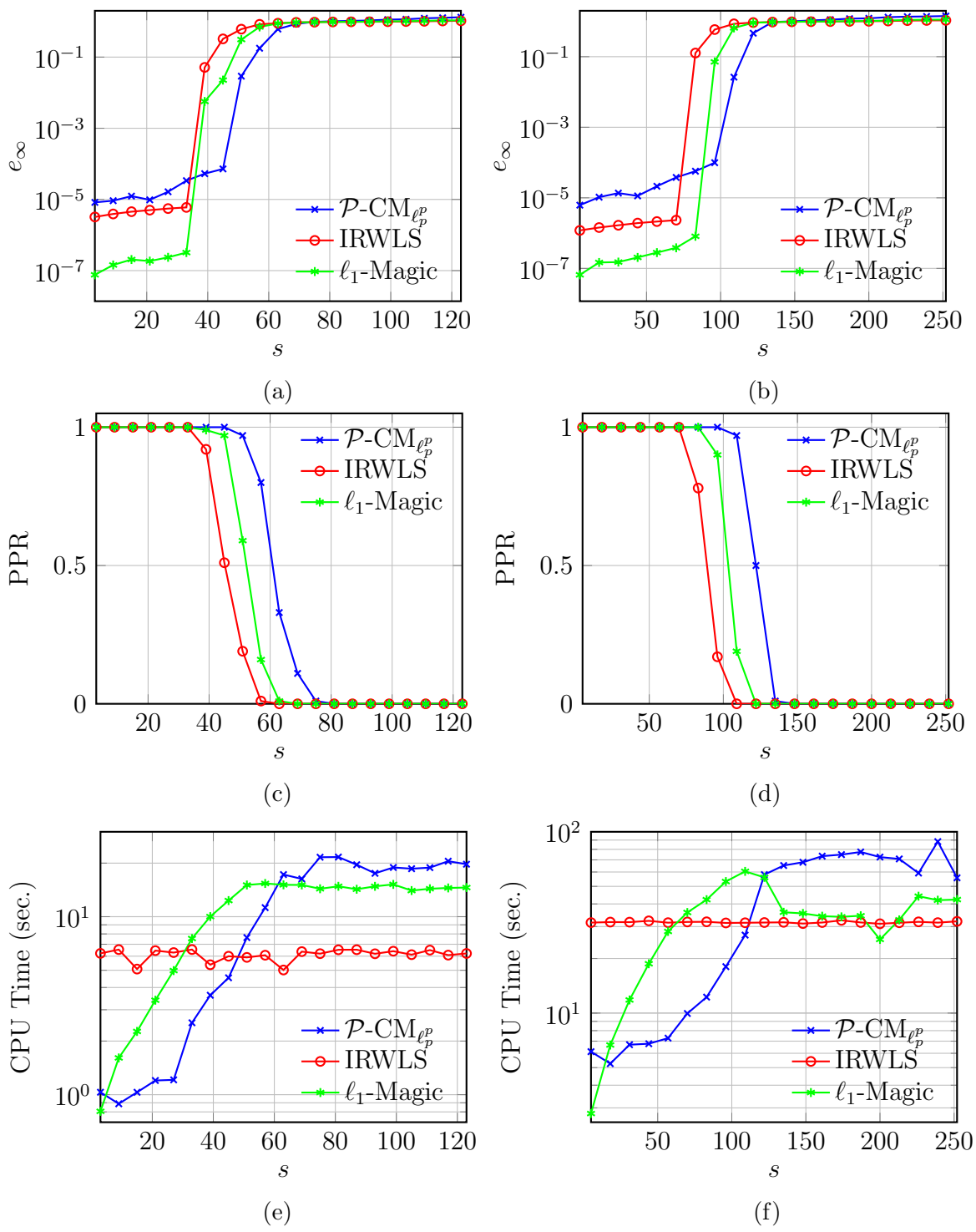


Figure 3.3: RP and CC of $\mathcal{P}$-class and competing methods for noiseless signals: (a) Average ℓ_∞ recovery error for $n = 2,048$, (b) Average ℓ_∞ recovery error for $n = 4,096$, (c) PPR for $n = 2,048$, (d) PPR for $n = 4,096$, (e) Average CPU time for $n = 2,048$, and (f) Average CPU time for $n = 4,096$.

with a reduced CC relative to those of the competing methods in simulations where the sparse signal is always perfectly reconstructed. For instance, when $n = 2,048$ the average CPU time required by the $\mathcal{P}$-CM$_{\ell_p^p}$, IRWLS, and ℓ_1-Magic methods were of 2.0, 6.1, and 5.4 seconds, respectively, for $s \leq 45$. On the other hand, when $n = 4,096$ the average CPU time required by the $\mathcal{P}$-CM$_{\ell_p^p}$, IRWLS, and ℓ_1-Magic methods were of 9.0, 31.7, and 24.9 seconds, respectively, for $s \leq 96$. However, the CC of the proposed method is increased relative to that of the competing methods for simulations where the sparse signals was not always perfectly recovered.

The MC of signals recovered by the $\mathcal{P}$-CM$_{\ell_p^p}$ method is compared with those of the IRWLS and ℓ_1-Magic methods in Figs. 3.4 and 3.5. Here MC is measured in terms of how close $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ is to the target value of $\delta = 0$ (see Fig. 1.4). As can be seen in the box plots[2] of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ in Fig. 3.4, signals recovered with the $\mathcal{P}$-CM$_{\ell_p^p}$ method are more consistent with the measurements taken than those recovered with the IRLWS and ℓ_1-Magic methods when $n = 2,048$. For instance, for simulations where $s = 45$ the median, and the minimum and maximum observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the $\mathcal{P}$-CM$_{\ell_p^p}$ method were 3.73×10^{-15}, 2.88×10^{-15}, and 4.34×10^{-15}, respectively, as shown in Fig. 3.4a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the IRWLS method for the same simulations were 3.8×10^{-15}, 2.58×10^{-15}, and 2.3×10^{-14} as shown in Fig. 3.4b while those obtained with the ℓ_1-Magic method were 1.85×10^{-8}, 9.75×10^{-9}, and 7.58×10^{-8} as shown in Fig. 3.4c. As can be seen in the box plots of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ in Fig. 3.5, signals recovered with the $\mathcal{P}$-CM$_{\ell_p^p}$ method are also more consistent with the measurements taken than those recovered with the IRLWS and ℓ_1-Magic methods when $n = 4,096$. For instance, for simulations where $s = 96$ the median, and the minimum and maximum observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the $\mathcal{P}$-CM$_{\ell_p^p}$ method were 7.53×10^{-15}, 6.73×10^{-15}, and 8.81×10^{-15}, respectively, as shown in Fig. 3.5a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the IRWLS method for the same simulations were 3.43×10^{-14}, 2.18×10^{-14}, and 6.12×10^{-14} as shown in Fig. 3.5b while those obtained with the ℓ_1-Magic method were 9.69×10^{-8}, 1.62×10^{-8}, and 4.59×10^{-7} as shown in Fig. 3.5c.

Results for noisy signals and orthogonal ensembles

RP and CC of the $\mathcal{P}$-CM$_{\text{ln}}$ method are compared with those of the DC$_{\text{ln}}$ and SPGL1 methods in Fig. 3.6. As can be seen in Figs. 3.6a to 3.6d, the proposed method

[2] A brief explanation of box plots can be found on p. 19. See [77] for a detailed description.

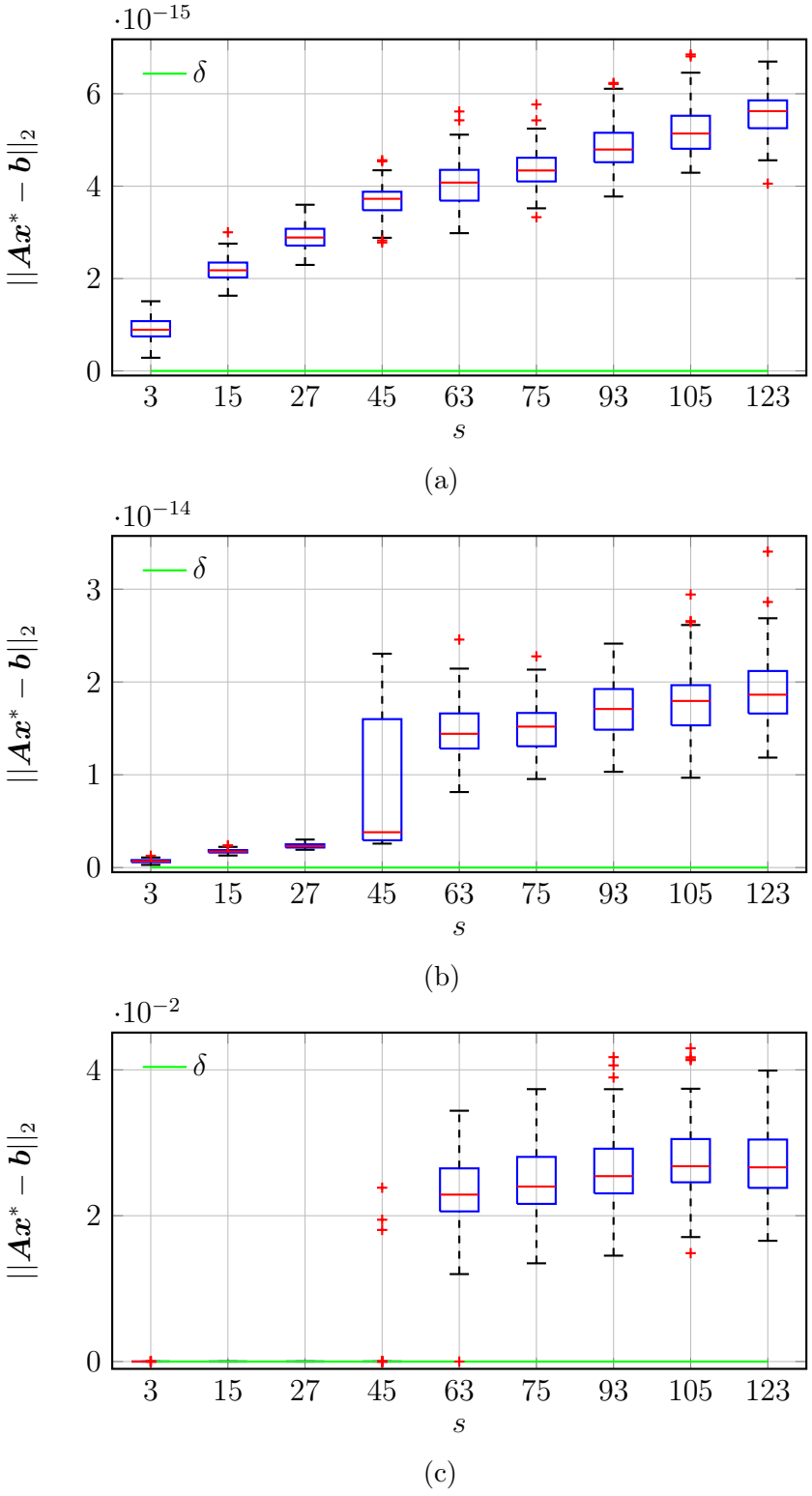


Figure 3.4: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ for noiseless signals of $n = 2,048$: (a) $\mathcal{P}$-CM$_{\ell_p^p}$ method, (b) IRWLS method, and (c) ℓ_1-Magic method.

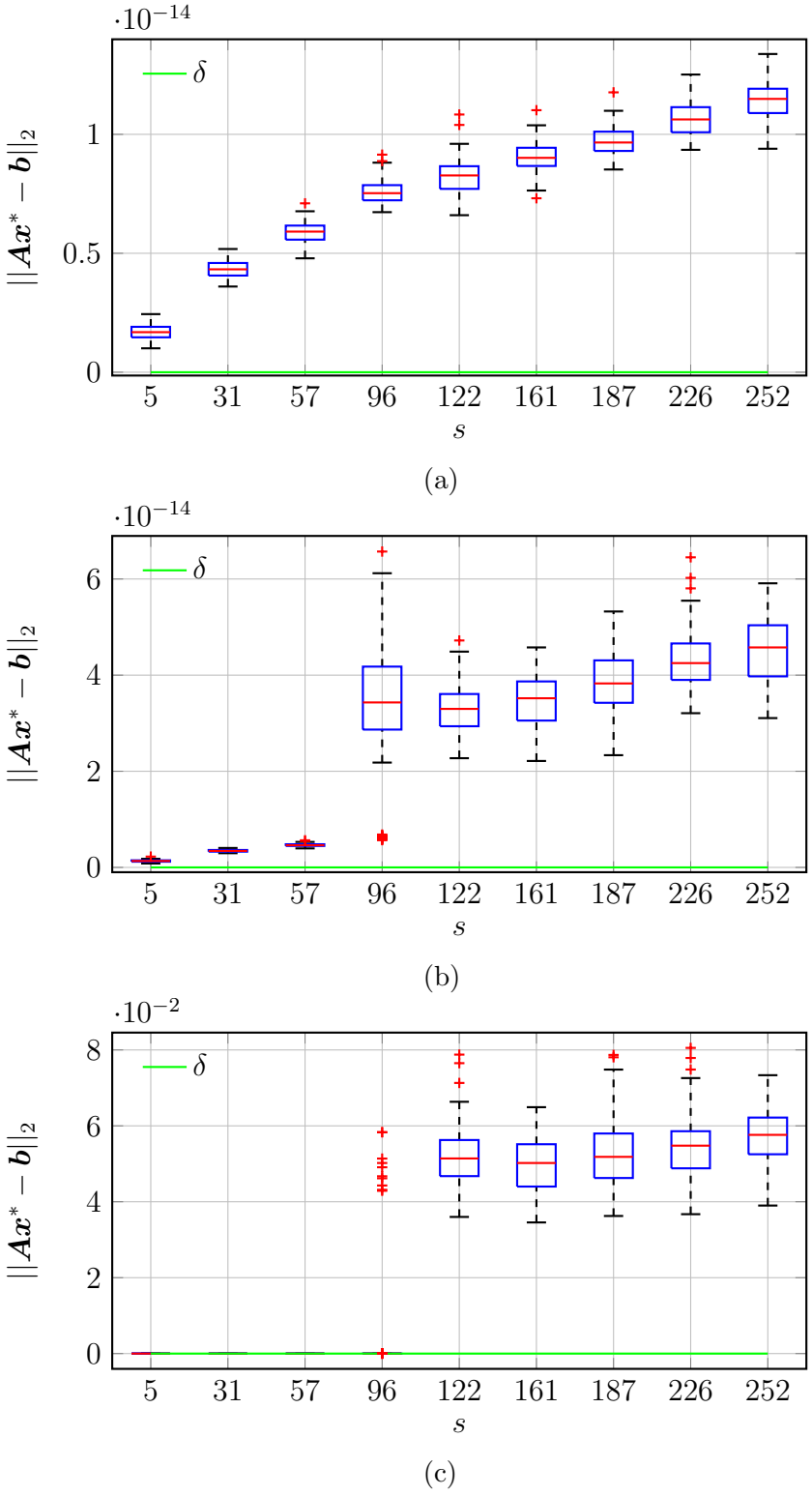


Figure 3.5: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ for noiseless signals of $n = 4,096$: (a) $\mathcal{P}$-CM$_{\ell_p^p}$ method, (b) IRWLS method, and (c) ℓ_1-Magic method.

achieved superior RP relative to that of the competing methods for different signal sizes. For instance, when $n = 16,384$ the average ℓ_∞ reconstruction error of the $\mathcal{P}$-CM$_{\text{ln}}$, DC$_{\text{ln}}$, and SPGL1 methods in Fig. 3.6a were 0.003, 1.53, and 1.69, respectively, for $s \leq 632$ while the MRF for perfect reconstruction in Fig. 3.6c has dropped from $2,048/224 \approx 9.1$ and $2,048/428 \approx 4.8$ in the SPGL1 and DC$_{\text{ln}}$ methods, respectively, to $2,048/632 \approx 3.2$ in the $\mathcal{P}$-CM$_{\text{ln}}$ method. On the other hand, when $n = 32,768$ the average ℓ_∞ reconstruction error of the $\mathcal{P}$-CM$_{\text{ln}}$, DC$_{\text{ln}}$, and SPGL1 methods in Fig. 3.6d were 0.0032, 1.53, 1.77, respectively, for $s \leq 1,265$ while the MRF for perfect reconstruction in Fig. 3.6d has dropped from $4,096/551 \approx 7.4$ and $4,096/857 \approx 4.8$ in the SPGL1 and DC$_{\text{ln}}$ methods, respectively, to $4,096/1,265 \approx 3.2$ in the $\mathcal{P}$-CM$_{\text{ln}}$ method. As can be seen in Figs. 3.6e and 3.6f, the superior RP of the proposed method comes with a CC that is comparable to that of SPGL1 method and reduced to that of DC$_{\text{ln}}$ method in simulations where the sparse signal is always perfectly reconstructed. For instance, when $n = 16,384$ the average CPU time required by the $\mathcal{P}$-CM$_{\text{ln}}$, DC$_{\text{ln}}$, and SPGL1 methods were of 10.2, 195.5, and 7.2 seconds, respectively, for $s \leq 632$. On the other hand, when $n = 32,768$ the average CPU time required by the $\mathcal{P}$-CM$_{\text{ln}}$, DC$_{\text{ln}}$, and SPGL1 methods were of 19.2, 318.1, and 13.7 seconds, respectively, for $s \leq 1,265$. However, the CC of the proposed method is increased relative to that of the SPGL1 method and comparable relative to that of the DC$_{\text{ln}}$ method for simulations where the sparse signals was not always perfectly recovered.

The consistency of signals recovered by the $\mathcal{P}$-CM$_{\text{ln}}$ method with respect to measurements taken is compared with those of the DC$_{\text{ln}}$ and SPGL1 methods in Figs. 3.7 and 3.8. Here consistency is measured in terms of how close $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ is to the target values of $\delta = 45.25 \times 10^{-4}$ and $\delta = 64 \times 10^{-4}$ for $n = 16,384$ and $n = 32,768$, respectively (see Fig. 1.4). As can be seen in the box plots of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ in Fig. 3.7, signals recovered with the $\mathcal{P}$-CM$_{\text{ln}}$ method are more consistent with the measurements taken than those recovered with the DC$_{\text{ln}}$ and SPGL1 methods when $n = 16,384$. For instance, for simulations where $s = 632$ the median, and the minimum and maximum observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the $\mathcal{P}$-CM$_{\text{ln}}$ method were 45.22×10^{-4}, 45.21×10^{-4}, and 45.23×10^{-4}, respectively, as shown in Fig. 3.7a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the DC$_{\text{ln}}$ method for the same simulations were 73.61×10^{-4}, 62.87×10^{-4}, and 91.21×10^{-4} as shown in Fig. 3.7b while those obtained with the SPGL1 method were 46.19×10^{-4}, 45.8×10^{-4}, and 46.25×10^{-4} as shown in Fig. 3.7c. As can be seen in the box plots of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ in Fig. 3.8, signals recovered with the $\mathcal{P}$-CM$_{\text{ln}}$ method are also more consistent with

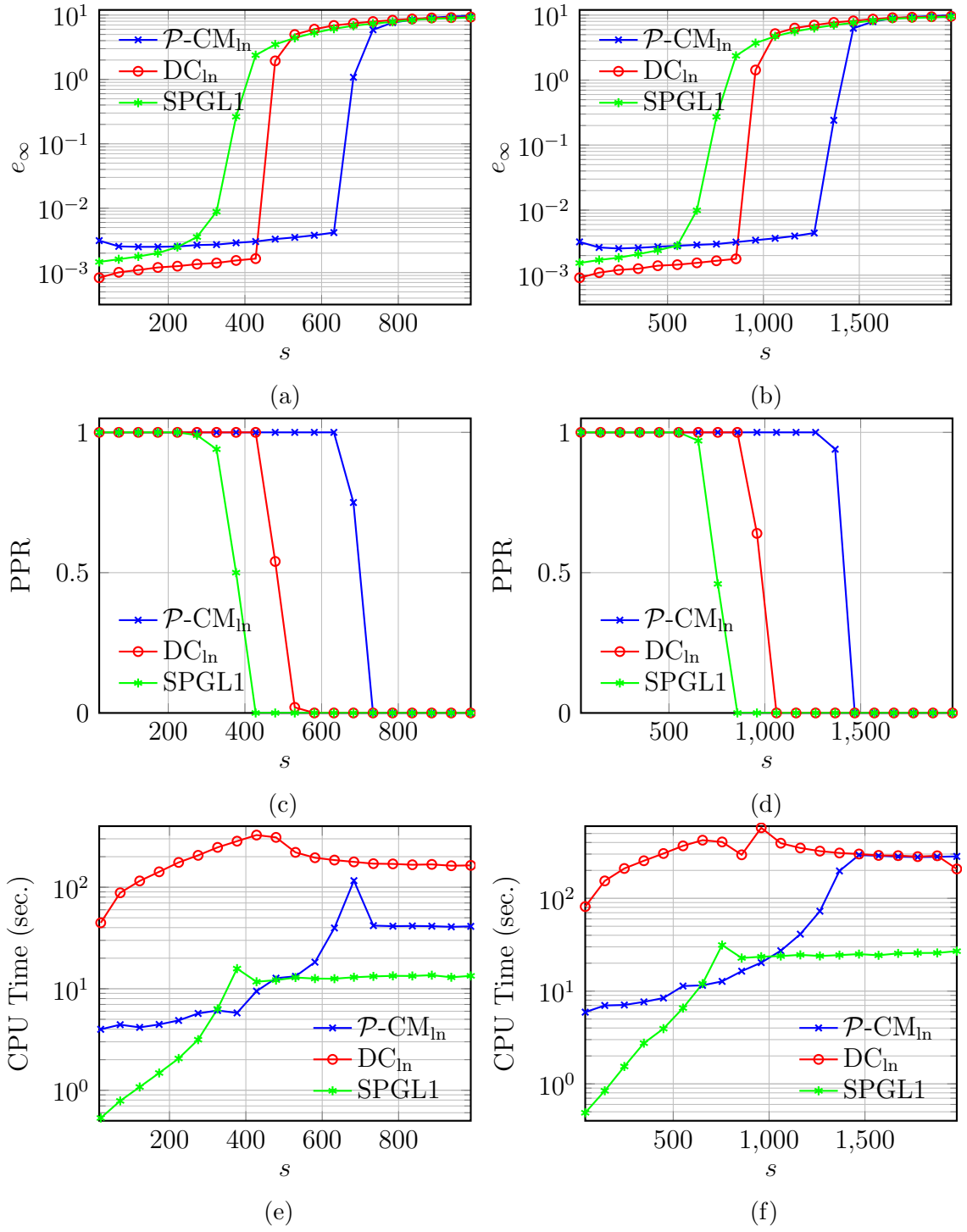


Figure 3.6: RP and CC of $\mathcal{P}$-class and competing methods for noisy signals: (a) Average ℓ_∞ recovery error for $n = 16,384$, (b) Average ℓ_∞ recovery error for $n = 32,768$, (c) PPR for $n = 16,384$, (d) PPR for $n = 32,768$, (e) Average CPU time for $n = 16,384$, and (f) Average CPU time for $n = 32,768$.

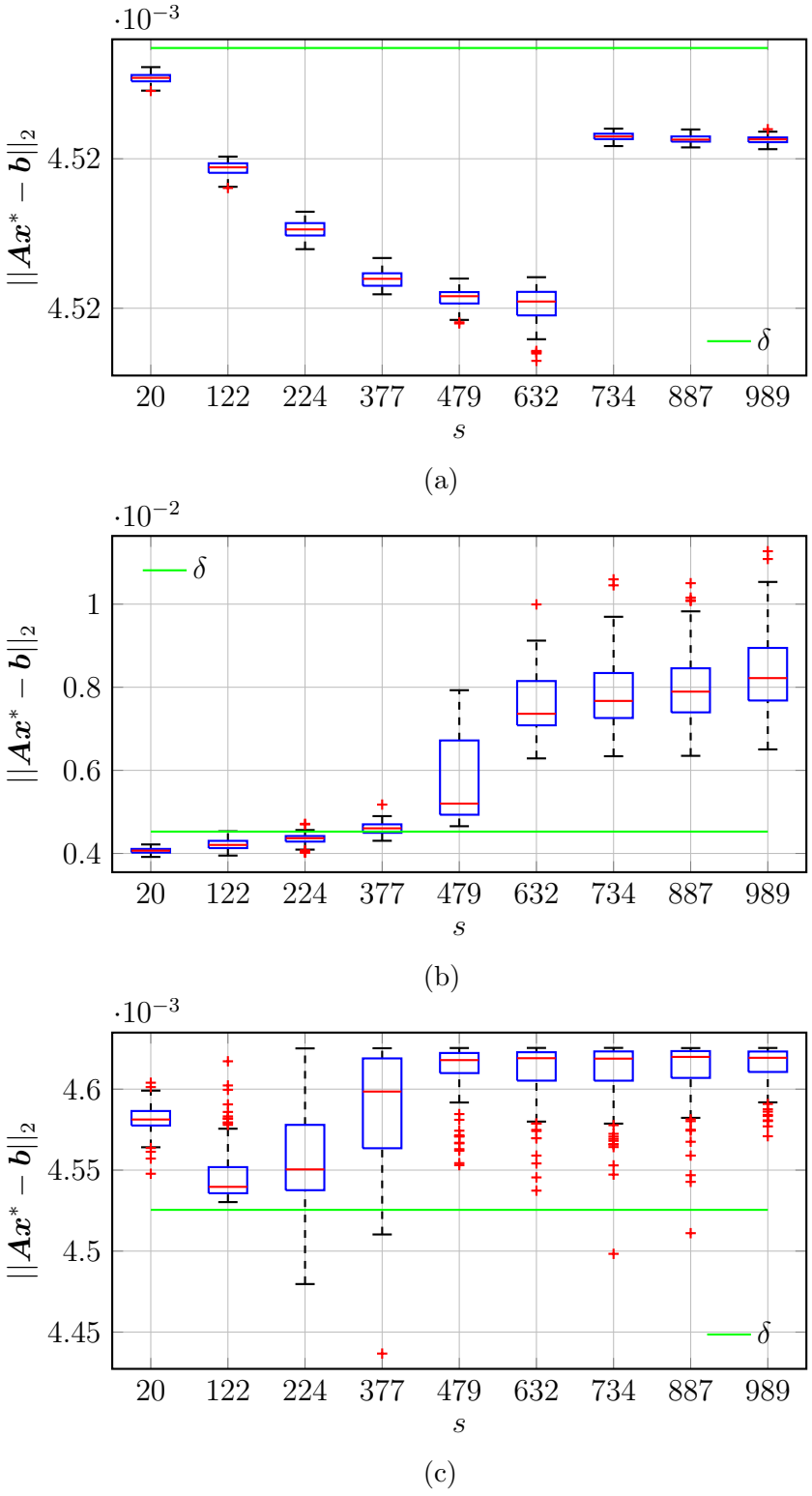
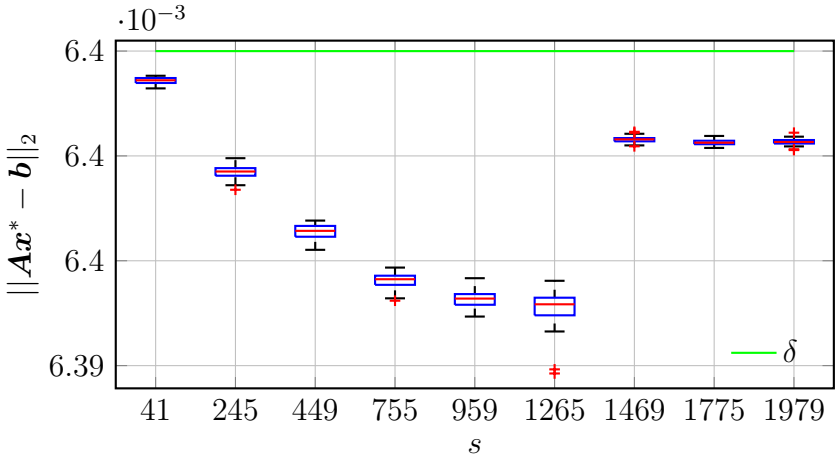
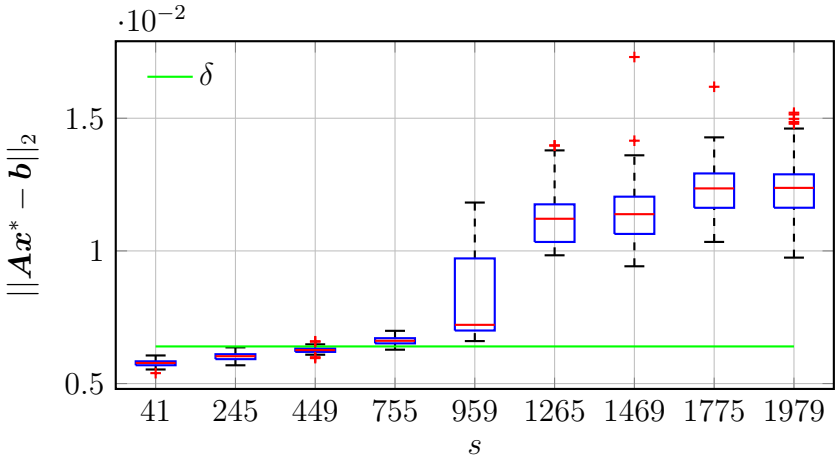


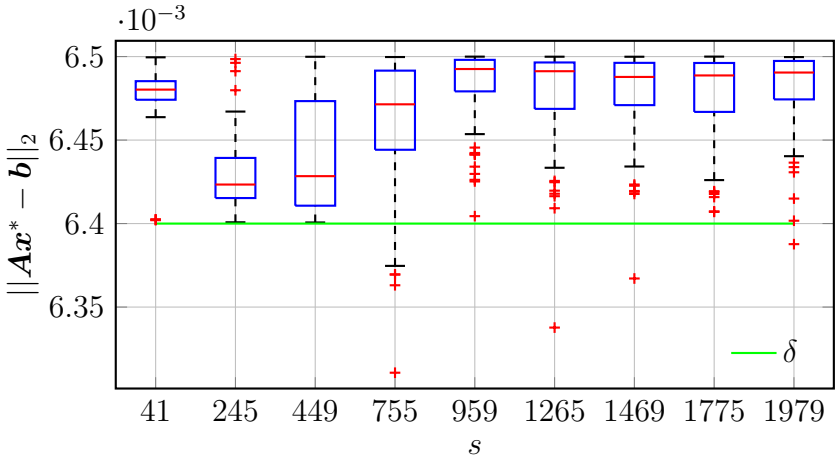
Figure 3.7: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ for noisy signals of $n = 16,384$: (a) $\mathcal{P}$-CM_{ln} method, (b) DC_{ln} method, and (c) SPGL1 method.



(a)



(b)



(c)

Figure 3.8: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ for noisy signals of $n = 32,768$: (a) $\mathcal{P}$-CM$_{\ln}$ method, (b) DC$_{\ln}$ method, and (c) SPGL1 method.

the measurements taken than those recovered with the DC_{ln} and SPGL1 methods when $n = 32,768$. For instance, for simulations where $s = 1,265$ the median, and the minimum and maximum observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the $\mathcal{P}$-CM_{ln} method were 63.95×10^{-4}, 63.94×10^{-4}, and 63.96×10^{-4}, respectively, as shown in Fig. 3.8a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ obtained with the DC_{ln} method for the same simulations were 112.12×10^{-4}, 98.34×10^{-4}, and 137.88×10^{-4} as shown in Fig. 3.8b while those obtained with the SPGL1 method were 64.91×10^{-4}, 64.33×10^{-4}, and 64.99×10^{-4} as shown in Fig. 3.8c.

3.4.3 Scalability of proposed family of BP methods

We carried out recovery simulations using orthogonal ensembles to evaluate the scalability of the proposed and competing methods for the recovery of noisy signals. The effect of problem size on RP and CC of the $\mathcal{P}$-$\text{CM}_{\ell_p^p}$ and SPGL1 methods were assessed for several values of n in the range of $2,048$ to $262,144$. The results are plotted in Fig. 3.9. As can be seen in Fig. 3.9a, the MRFs obtained with the $\mathcal{P}$-$\text{CM}_{\ell_p^p}$ method are significantly smaller than those obtained with the SPGL1 method from small- to very-large-scale reconstruction problems.

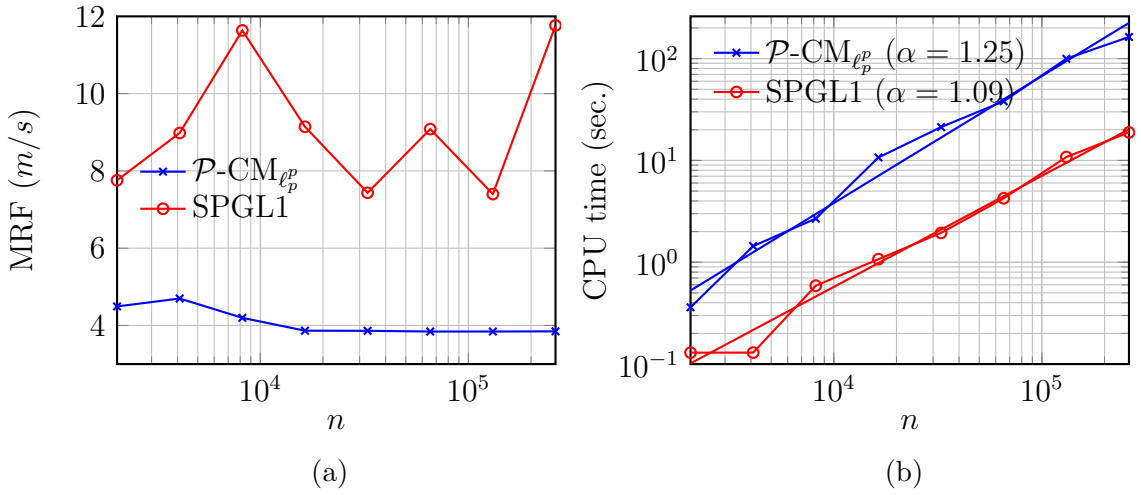


Figure 3.9: Scalability assessment of $\mathcal{P}$-class and competing methods: (a) MRF for perfect reconstruction and (b) Average CPU time.

We assumed during the recovery simulations that the CC was of order n^α and obtained empirical estimates of the exponent α as was done in [43]. The average CPU time for the $\mathcal{P}$-$\text{CM}_{\ell_p^p}$ and SPGL1 methods is plotted in Fig. 3.9b. As can be

seen, the empirical exponent values obtained with the $\mathcal{P}$-CM$_{\ell_p^p}$ method are close to those obtained with the SPGL1 method. In addition, both methods have an empirical complexity less than quadratic.

The effect of problem size on the convergence rate of sequence $\left\{P_\epsilon(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ was assessed for several values of n in the range of $2,048$ to $262,144$. The results are plotted in Fig. 3.10 for simulations where the signal is always perfectly recovered. As can be seen, an average of 3.5 to 5.4 points are computed for small- to very-large-scale reconstruction problems.

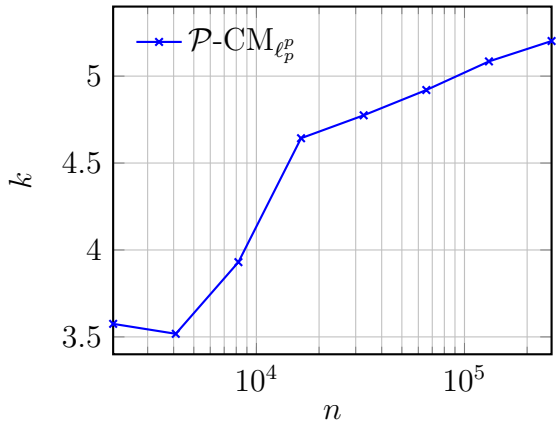


Figure 3.10: Scalability assessment of the convergence rate.

3.5 Conclusions

A new family of signal-recovery methods has been described. Sparsity is promoted with the class $\mathcal{P}$ of SPFs which (1) is fairly general as it includes some widely used SPFs of class $\mathcal{N}$ such as those in (1.4) to (1.6), and (2) it leads to an efficient and robust recovery process. Results obtained pertaining to the optimality conditions of $\mathcal{P}$-class problems show that (1) stationary points with at most m nonzero coordinates are local minimizers, and (2) a global minimizer is a stationary point with at most m nonzero coordinates.

In the new family of SCF methods, the solution of a $\mathcal{P}$-class problem is approached by employing a PLA of the $\mathcal{P}$-class function. Convex subproblems are formulated as weighted ℓ_1-norm minimization problems while an efficient FOS based on NESTA [9] is employed. Thus, the proposed solver is applicable for the recovery of large signals from Gaussian or orthogonal ensembles. The sequence of solution points was shown

to be a monotonically decreasing sequence of values of the objective function and converges to a local minimizer with at most m nonzero coordinates.

Simulation results demonstrate that the new methods are robust, lead to fast convergence, and achieve superior RP metrics in terms of increased PPRs, reduced MRFs for perfect recovery, and reduced average ℓ_∞ reconstruction error when compared with the ℓ_1-Magic [18], IRWLS [37], SPGL1 [11], and DC-family [46] methods. CC metrics in terms of the average CPU time of the new methods were found to be comparable to that of the SPGL1 method and reduced to those of the ℓ_1-Magic, IRWLS, and DC-family methods. In addition, scalability results demonstrated that the new methods are well suited for large-scale recovery problems.

Chapter 4

A New Proximal-Point Based Method

4.1 Introduction

State-of-the-art methods applicable to nonconvex recovery problems [23, 27, 46, 104, 105] are based on an indirect solution approach where approximation is employed (see Sec. 1.2.3 for details). Unfortunately, indirect methods are inefficient for the solution of very-large-scale problems, typically in the range of a million variables, as they entail the solution of several subproblems of the same scale in sequence.

In this chapter, a new proximal-point (PP) based method that solves very-large-scale nonconvex optimization problems is proposed. Sparse-signal recovery is carried out by minimizing the sum of two functions, namely, the indicator of a closed ball under an affine mapping and the ϵ-ℓ_p^p norm functions. The objective function obtained in this way exhibits unusually rich properties from an optimization perspective. A PP method based on the update-formula in (1.14) is used for minimizing the objective function and a continuation procedure is employed so that a minimum can be found efficiently for arbitrarily small values of ϵ. When the Moreau envelope (ME) in (1.13) is computed approximately, the update-formula can be applied by iteratively performing two fundamental operations, namely, (1) computation of the PP of the ϵ-ℓ_p^p norm function and (2) projection of the PP onto the closed ball under affine mapping. The first operation can be performed either analytically or numerically by using a fast iterative method when p can be expressed by a common fraction. The second operation is performed efficiently by computing a sequence of closed-form

projectors onto convex sets. The sequence of points associated with the iterative computation is shown to converge to a minimizer of the problem at hand and a two-step method with optimal convergence rate given by (1.21) is employed for accelerated convergence. Simulations carried out with the proposed method show that very-large signals can be recovered accurately and efficiently and that the solutions obtained are superior to those obtained with competing state-of-the-art methods while requiring a comparable amount of computation.

The chapter is organized as follows. In Sec. 4.2, the proposed recovery problem is described and its feasible set and objective function are examined. In Sec. 4.3, the new recovery method is described. In Sec. 4.4, simulation results for the proposed and corresponding competing methods are presented. In Sec. 4.5, conclusions are drawn.

4.2 Proposed Recovery Problem

Hereafter, $P_\epsilon(\boldsymbol{x})$ is given by (1.2) where $\boldsymbol{w}$ is a column vector of n ones and $p_\epsilon(|x_i|)$ is defined by (1.4), and column vector $\boldsymbol{g} \in \boldsymbol{\partial} P_\epsilon(\boldsymbol{x})$ is a subgradient of $P_\epsilon(\boldsymbol{x})$ where $\boldsymbol{\partial} P_\epsilon(\boldsymbol{x})$ is the subdifferential as in Definition 8.3 of [96]. Here we introduce an unconstrained reformulation of the problem in (1.7) that can be solved with PP methods.

Let us rewrite problem (BP_δ) in (1.7) in the compact form

$$\underset{\boldsymbol{x}\in\mathcal{K}_\delta}{\text{minimize}} \quad P_\epsilon(\boldsymbol{x}) \tag{4.1}$$

where feasible set $\mathcal{K}_\delta$ denotes the closed ball in $\mathbb{R}^n$ under an affine mapping given by

$$\mathcal{K}_\delta = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\| \leq \delta\} \tag{4.2}$$

The minimization of function $P_\epsilon(\boldsymbol{x})$ over set $\mathcal{K}_\delta$ is equivalent to the minimization of the sum of function $P_\epsilon(\boldsymbol{x})$ and the indicator function of set $\mathcal{K}_\delta$ over all $\mathbb{R}^n$ (see p. 7 of [96]). Thus, problem (BP_δ) in (1.7) is equivalent to the unconstrained optimization problem

$$(\text{LP}_{\epsilon,\delta}) \qquad \underset{\boldsymbol{x}}{\text{minimize}}\ F_{\epsilon,\delta}(\boldsymbol{x}) \tag{4.3}$$

where $F_{\epsilon,\delta}(\boldsymbol{x})$ is a function of the form

$$F_{\epsilon,\delta}(\boldsymbol{x}) = P_\epsilon(\boldsymbol{x}) + I_{\mathcal{K}_\delta}(\boldsymbol{x}) \tag{4.4}$$

and $I_{\mathcal{K}_\delta}(\boldsymbol{x})$ is the indicator function of set $\mathcal{K}_\delta$ given by

$$I_{\mathcal{K}_\delta}(\boldsymbol{x}) = \begin{cases} 0 & \boldsymbol{x} \in \mathcal{K}_\delta \\ +\infty & \boldsymbol{x} \notin \mathcal{K}_\delta \end{cases} \tag{4.5}$$

In Sec. 4.2.1, we show that set $\mathcal{K}_\delta$ can be expressed as the intersection of a finite number of closed and convex sets which facilitates the computation of its projector. In Sec. 4.2.2, we present several properties of function $P_\epsilon(\boldsymbol{x})$ that are applicable to the computation of its PP mapping. In Sec. 4.2.3, we define the solution set of problem $(\text{LP}_{\epsilon,\delta})$ and introduce a sequence of values of the regularization parameter ϵ. The sequence is applicable to the solution of problem $(\text{LP}_{\epsilon,\delta})$ when ϵ approaches 0. In Sec. 4.2.4, we present several properties of the subdifferential mapping and ME of function $F_{\epsilon,\delta}(\boldsymbol{x})$ in (4.4). These properties are related to the convergence of PP methods for the problem at hand.

4.2.1 Feasible Set

Let

$$\boldsymbol{A}\boldsymbol{z} = \boldsymbol{c} + \boldsymbol{b} \tag{4.6}$$

where $\boldsymbol{z} \in \mathbb{R}^n$ and $\boldsymbol{c} \in \mathbb{R}^m$. Combining (4.2) and (4.6), we conclude that the feasible set $\mathcal{K}_\delta$ can be written as

$$\mathcal{K}_\delta = \{(\boldsymbol{z}, \boldsymbol{c}) \in \mathbb{R}^n \times \mathbb{R}^m : \boldsymbol{c} = \boldsymbol{A}\boldsymbol{z} - \boldsymbol{b},\ ||\boldsymbol{c}|| \leq \delta\} \tag{4.7}$$

The linear system of equations in (4.7) is related to the hyperplanes given by

$$\mathcal{K}_{\delta,i} = \left\{(\boldsymbol{z}, \boldsymbol{c}) \in \mathbb{R}^n \times \mathbb{R}^m : \boldsymbol{a}_i^T \boldsymbol{z} - \mathbf{1}_i^T \boldsymbol{c} = b_i\right\}, \qquad \text{for } i \in \mathcal{I}_m \tag{4.8}$$

where $\mathbf{1}_i$ is a column vector of length m with all coordinates zero valued with the exception of the ith coordinate which assumes the value of 1 and $\mathcal{I}_m = \{1,\ \ldots,\ m\}$.

The hyperplanes in (4.8) have an equivalent algebraic definition in terms of linear varieties as stated in the following proposition.

Proposition 4.1 (Linear Varieties)**.** *Each set $\mathcal{K}_{\delta,i}$ in (4.8) is a linear variety.*

Proof. Each set $\mathcal{K}_{\delta,i}$ in (4.8) defines a hyperplane in $\mathbb{R}^{(m+1)}$ (see Proposition 2 of [74]) and a hyperplane in $\mathbb{R}^{(m+1)}$ is an m-dimensional linear variety (see definition on p. 517

of [74]). Therefore, each set $\mathcal{K}_{\delta,i}$ in (4.8) is a linear variety. □

The closed ball in $\mathbb{R}^m$ in (4.7) is related to the closed ball in $\mathbb{R}^{(n+m)}$ under an affine mapping given by

$$\mathcal{K}_{\delta,i} = \left\{ (\boldsymbol{z}, \boldsymbol{c}) \in \mathbb{R}^n \times \mathbb{R}^m : \left\| \begin{bmatrix} \mathbf{0}^{n\times n} & \mathbf{0}^{n\times m} \\ \mathbf{0}^{m\times n} & \boldsymbol{I}^{m\times m} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{c} \end{bmatrix} \right\| \leq \delta \right\}, \quad \text{for } i \in \{m+1\} \tag{4.9}$$

where $\mathbf{0}^{n\times n}$, $\mathbf{0}^{n\times m}$, and $\mathbf{0}^{m\times n}$ are $n \times n$, $n \times m$, and $m \times n$ zero matrices, respectively, and $\boldsymbol{I}^{m\times m}$ is the $m \times m$ identity matrix. Combining (4.7) to (4.9), we conclude that set $\mathcal{K}_\delta$ in (4.2) can be expressed as

$$\mathcal{K}_\delta = \bigcap_{i=1}^{m+1} \mathcal{K}_{\delta,i} \tag{4.10}$$

Expressing the feasible set as the intersection of closed and convex sets, as in (4.10), facilitates the computation of the projector onto the feasible set. Efficient methods for computing a point in the intersection of convex sets, such as the alternating projection (AP) method [52], are also applicable for computing the projector onto the intersection of these sets (see Corollary 2 of [114]).

4.2.2 Sparsity Promoting Function

Let $\mathbb{R}^+$ and $\mathbb{R}^-$ denote the set of all positive and negative real numbers, respectively. From the definition of the absolute value of x_i, function $p_\epsilon(|x_i|)$ can be written as

$$p_\epsilon(|x_i|) = \begin{cases} p_{\epsilon,-}(x_i), & x_i \in \mathbb{R}^- \\ p_{\epsilon,+}(x_i), & x_i \in \mathbb{R}^+ \\ \epsilon^p, & x_i = 0 \end{cases} \tag{4.11}$$

where

$$p_{\epsilon,-}(x_i) = (-x_i + \epsilon)^p \tag{4.12a}$$

and

$$p_{\epsilon,+}(x_i) = (x_i + \epsilon)^p \tag{4.12b}$$

On the basis of (1.18), (1.23), and (4.11), the following proposition states that the sparsity-promoting function (SPF) at hand possess two important properties since

(1) the magnitude of its subgradients is upper bounded and (2) it can be expressed as the difference between a finite convex function and a positive multiple of $\frac{1}{2}|x_i|^2$.

Proposition 4.2 (Lipschitz Continuity and Lower-C^2 Properties at $\mathbb{R}$)**.** *Function $p_\epsilon(|x_i|)$ is Lipschitz continuous with constant*

$$\kappa_\epsilon = \frac{p}{\epsilon^{1-p}} \tag{4.13}$$

and lower-C^2 with constant

$$\rho_\epsilon = \frac{p\,|p-1|}{\epsilon^{2-p}} \tag{4.14}$$

at every $x_i \in \mathbb{R}$.

Proof. The first-order derivatives of functions $p_{\epsilon,-}(x_i)$ and $p_{\epsilon,+}(x_i)$ in 4.12a and (4.12b) are given, respectively, by

$$p'_{\epsilon,-}(x_i) = -p(-x_i+\epsilon)^{p-1}$$

and

$$p'_{\epsilon,+}(x_i) = p(x_i+\epsilon)^{p-1}$$

Because $p'_{\epsilon,-}(x_i)$ and $p'_{\epsilon,+}(x_i)$ are non-increasing for increasing values of x_i, it can be shown that they are bounded as

$$0 < \left|p'_{\epsilon,-}(x_i)\right| \leq \frac{p}{\epsilon^{1-p}} \quad \text{and} \quad 0 < \left|p'_{\epsilon,+}(x_i)\right| \leq \frac{p}{\epsilon^{1-p}} \tag{4.15}$$

and from (1.17), $p_{\epsilon,-}(x_i)$ and $p_{\epsilon,+}(x_i)$ are Lipschitz continuous functions with constant κ_ϵ as described in (4.13). Furthermore, since

$$\lim_{x_i\to 0^-} p_\epsilon(|x_i|) = \lim_{x_i\to 0^+} p_\epsilon(|x_i|) = \epsilon^p$$

the limit $\lim\limits_{x_i\to 0} p_\epsilon(|x_i|)$ exists and

$$\lim_{x_i\to \tilde{x}_i} p_\epsilon(|x_i|) = p_\epsilon(|\tilde{x}_i|), \qquad \text{for } \tilde{x}_i = 0 \tag{4.16}$$

Consequently, $p_\epsilon(|x_i|)$ is continuous at $x_i = 0$ and, therefore, Lipschitz continuous with constant κ_ϵ by virtue of (4.11), (4.15), and (4.16).

Now let $h(|x_i|)$ denote a function of the form

$$h(|x_i|) = p_\epsilon(|x_i|) + \frac{1}{2}\rho|x_i|^2 \tag{4.17}$$

where $\rho > 0$. Using an approach similar to that used for $p_\epsilon(|x_i|)$, we have

$$h(|x_i|) = \begin{cases} h_-(x_i), & x_i \in \mathbb{R}^- \\ h_+(x_i), & x_i \in \mathbb{R}^+ \\ \epsilon^p, & x_i = 0 \end{cases} \tag{4.18}$$

where

$$h_-(x_i) = p_{\epsilon,-}(x_i) + \frac{1}{2}\rho(-x_i)^2$$

and

$$h_+(x_i) = p_{\epsilon,+}(x_i) + \frac{1}{2}\rho(x_i)^2$$

The second-order derivatives of $p_{\epsilon,-}(x_i)$ and $p_{\epsilon,+}(x_i)$ have the minimum and maximum values

$$\min_{x_i\in\mathbb{R}^-} p''_{\epsilon,-}(x_i) = \lim_{x_i\to 0^-} p''_{\epsilon,-}(x_i) = \min_{x_i\in\mathbb{R}^+} p''_{\epsilon,+}(x_i) = \lim_{x_i\to 0^+} p''_{\epsilon,+}(x_i) = \frac{p(p-1)}{\epsilon^{2-p}} \tag{4.19}$$

and

$$\max_{x_i\in\mathbb{R}^-} p''_{\epsilon,-}(x_i) = \lim_{x_i\to -\infty} p''_{\epsilon,-}(x_i) = \max_{x_i\in\mathbb{R}^+} p''_{\epsilon,+}(x_i) = \lim_{x_i\to +\infty} p''_{\epsilon,+}(x_i) = 0 \tag{4.20}$$

respectively. Based on (4.19) and (4.20), we conclude that $h''_-(x_i) \geq 0$ and $h''_+(x_i) \geq 0$ for values of ρ greater than or equal to that given in (4.14). Thus, $h_-(x_i)$ and $h_+(x_i)$ are convex functions for ρ as given in (4.14).

Lastly, since $h(|x_i|)$ is a continuous function at $x_i = 0$, and $h_-(x_i)$ and $h_+(x_i)$ are convex functions, we conclude from (4.18) that $h(|x_i|)$ must be a convex function for ρ as given in (4.14). Thus, $p_\epsilon(|x_i|)$ is a lower-C^2 function with constant ρ_ϵ at every $x_i \in \mathbb{R}$ because there is an expression of the form

$$p_\epsilon(|x_i|) = h(|x_i|) - \frac{1}{2}\rho_\epsilon|x_i|^2$$

where $h(|x_i|)$ is a convex function (see (1.23)). □

The properties of $p_\epsilon(|x_i|)$ in Proposition 4.2 can be easily extended to $P_\epsilon(\boldsymbol{x})$ as

stated in the following corollary.

Corollary 4.1 (Lipschitz Continuity and Lower-C^2 Properties at $\mathbb{R}^n$)**.** *Function $P_\epsilon(\boldsymbol{x})$ is Lipschitz continuous with constant*

$$\kappa_\epsilon = n\frac{p}{\epsilon^{1-p}} \tag{4.21}$$

and lower-C^2 at every $\boldsymbol{x} \in \mathbb{R}^n$ with constant ρ_ϵ given by (4.14).

Proof. From Proposition 4.2, function $p_\epsilon(|x_i|)$ is Lipschitz continuous with constant κ_ϵ as given in (4.21) and lower-C^2 at every $x_i \in \mathbb{R}$ with constant ρ_ϵ as given in (4.14). Thus, based on (1.2), we conclude that $P_\epsilon(\boldsymbol{x})$ must be Lipschitz continuous with constant κ_ϵ as given in (4.21) (see Theorem 12.1 of [41]) and lower-C^2 at every $\boldsymbol{x} \in \mathbb{R}^n$ with constant ρ_ϵ as given in (4.14) (see Exercise 10.35(a) of [96]). □

The properties in Corollary 4.1 and Proposition 4.2 are directly applicable to the computation of the PP of functions $p_\epsilon(|x_i|)$ and $P_\epsilon(\boldsymbol{x})$ because the PP mapping of lower-C^2, Lipschitz continuous functions is single valued [56].

4.2.3 Solution Set and Regularization Sequence

The normal cone of set $\mathcal{K}_\delta$, denoted by $N_{\mathcal{K}_\delta}$, is equivalent to the subdifferential of the indicator function $I_{\mathcal{K}_\delta}(\boldsymbol{x})$ (see Exercise 8.14 of [96]) and if we let $\boldsymbol{v} \in N_{\mathcal{K}_\delta}(\bar{\boldsymbol{x}})$ with $\bar{\boldsymbol{x}} \in \mathcal{K}_\delta$ denote a vector which is normal to set $\mathcal{K}_\delta$ at $\bar{\boldsymbol{x}}$, then we have the property

$$\boldsymbol{v}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \leq 0, \qquad \text{for } \boldsymbol{x} \in \mathcal{K}_\delta,\ \boldsymbol{v} \in N_{\mathcal{K}_\delta}(\bar{\boldsymbol{x}}) \tag{4.22}$$

(see Definition 6.3 of [96]). We define the solution set of problem ($\text{LP}_{\epsilon,\delta}$) as follows.

Definition 4.1 (Solution Set)**.** *Point $\boldsymbol{x}^*_{\epsilon,\delta}$ is a solution of problem ($LP_{\epsilon,\delta}$) if*

$$\boldsymbol{\partial} F_{\epsilon,\delta}(\boldsymbol{x}^*_{\epsilon,\delta}) \ni \mathbf{0} \iff \boldsymbol{\partial}\left[P_\epsilon(\boldsymbol{x}) + I_{\mathcal{K}_\delta}(\boldsymbol{x})\right] \ni \mathbf{0} \iff \boldsymbol{\partial} P_\epsilon(\boldsymbol{x}^*_{\epsilon,\delta}) + N_{\mathcal{K}_\delta}(\boldsymbol{x}^*_{\epsilon,\delta}) \ni \mathbf{0} \tag{4.23}$$

and there exists a closed ball of radius $\alpha \in (0, +\infty]$ centered at $\boldsymbol{x}^*_{\epsilon,\delta}$ given by

$$B(\alpha, \mathcal{S}_{\epsilon,\delta}) = \left\{\boldsymbol{x} : ||\boldsymbol{x} - \boldsymbol{x}^*_{\epsilon,\delta}|| \leq \alpha\right\} \tag{4.24}$$

such that

$$P_\epsilon(\boldsymbol{x}) \geq P_\epsilon(\boldsymbol{x}^*_{\epsilon,\delta}), \qquad \textit{for any } \boldsymbol{x} \in B(\alpha, \mathcal{S}_{\epsilon,\delta}) \tag{4.25}$$

Furthermore, the set of all points satisfying the conditions in (4.23) *and* (4.25) *defines the solution set of the problem* $(LP_{\epsilon,\delta})$*, denoted by* $\mathcal{S}_{\epsilon,\delta}$*.*

The distance between a given point and a solution point is defined as follows.

Definition 4.2 (Distance to Solution Set)**.** *The distance of a feasible point* $\boldsymbol{x} \in \mathcal{K}_\delta$ *to the solution set* $\mathcal{S}_{\epsilon,\delta}$*, denoted by* $d(\boldsymbol{x}, \mathcal{S}_{\epsilon,\delta})$*, is defined as*

$$d(\boldsymbol{x}, \mathcal{S}_{\epsilon,\delta}) = \inf_{\boldsymbol{x}^*_{\epsilon,\delta} \in \mathcal{S}_{\epsilon,\delta}} ||\boldsymbol{x} - \boldsymbol{x}^*_{\epsilon,\delta}|| \tag{4.26}$$

On the basis of Corollary 4.1, function $P_\epsilon(\boldsymbol{x})$ assumes large Lipchitz constants for arbitrarily small regularization values since $\kappa_\epsilon \to \infty$ as $\epsilon \to 0$ (see (4.21)). Finding a point of the solution set $\mathcal{S}_{\epsilon,\delta}$ in Definition 4.1 is challenging when the regularization parameter approaches zero because first-order solvers (FOSs), such as that in (1.14), exhibit slow convergence in the minimization of Lipschitz continuous functions with large Lipschitz constants [66]. Computation of such a solution point is facilitated when a sequence of problems of the form in (4.3) are solved for appropriate values of the regularization parameter. The following definition presents a sequence of regularization parameters.

Definition 4.3 (Regularization Sequence)**.** *Let* $\mathcal{I}_q$ *denote a set of* q *integers given by* $\mathcal{I}_q = \{1,\ 2,\ \ldots,\ q\}$ *and let* $\{\epsilon_j\}_{j \in \{0\} \cup \mathcal{I}_q}$ *denote a strictly decreasing sequence with the properties:*

1. ϵ_0 *is large enough so that, for given values of* n *and* p*, we have* $P_{\epsilon_0}(\boldsymbol{x}) - n\epsilon_0^p \approx 0$*, for all* $\boldsymbol{x} \in \mathcal{K}_\delta$*,*

2. ϵ_q *is approximately zero, and*

3. $\epsilon_{j-1} - \epsilon_j \leq \nu$*, for all* $j \in \mathcal{I}_q$ *where* ν *is an arbitrary small positive constant.*

Based on Definition 4.3, we define the sequence of problems $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j \in \{0\} \cup \mathcal{I}_q}$ of the form in (4.3) for different values of ϵ. The Euclidean distance between the solutions of two consecutive problems in such a sequence is described in the following lemma.

Lemma 4.1 (Distance Between Solutions)**.** *Let* $\{\epsilon_j\}_{j \in \mathcal{I}_q}$ *be a sequence as described in Definition 4.3, and let* $\mathcal{S}_{\epsilon_{j-1},\delta}$ *and* $\mathcal{S}_{\epsilon_j,\delta}$ *denote the solution sets of problems* $(LP_{\epsilon_{j-1},\delta})$ *and* $(LP_{\epsilon_j,\delta})$*, respectively. For any* $\boldsymbol{x}^*_{\epsilon_j,\delta} \in \mathcal{S}_{\epsilon_j,\delta}$ *and* $\boldsymbol{x}^*_{\epsilon_{j-1},\delta} \in \mathcal{S}_{\epsilon_{j-1},\delta}$*, we have*

$$0 \leq P_{\epsilon_{j-1}}(\boldsymbol{x}^*_{\epsilon_{j-1},\delta}) - P_{\epsilon_j}(\boldsymbol{x}^*_{\epsilon_j,\delta}) \leq n\nu^p, \qquad \text{for all } j \in \mathcal{I}_q \tag{4.27}$$

Furthermore, if the upper bound ν in Property 3 of Definition 4.3 is small enough such that

$$P_{\epsilon_{j-1}}(\boldsymbol{x}^*_{\epsilon_{j-1},\delta}) \approx P_{\epsilon_j}(\boldsymbol{x}^*_{\epsilon_{j-1},\delta}), \qquad \text{for all } j \in \mathcal{I}_q \tag{4.28}$$

then we have

$$||\boldsymbol{x}^*_{\epsilon_{j-1},\delta} - \boldsymbol{x}^*_{\epsilon_j,\delta}|| < \nu^p \frac{\epsilon_j^{1-p}}{p}, \qquad \text{for all } j \in \mathcal{I}_q \tag{4.29}$$

Proof. The binomial expansion of function $p_{\epsilon_j}(|x_i|)$ is given by

$$\begin{aligned} p_{\epsilon_j}(|x_i|) &= (|x_i| + \epsilon_j)^p \\ &= |x_i|^p + p|x_i|^{p-1}\epsilon_j + \frac{p(p-1)}{2!}|x_i|^{p-2}\epsilon_j^2 + \frac{p(p-1)(p-2)}{3!}|x_i|^{p-3}\epsilon_j^3 + \cdots \end{aligned}$$

(see binomial series, pp. 774 of [100]). From this expansion, we obtain

$$\lim_{|x_i|\to\infty} p_{\epsilon_j}(|x_i|) = \infty$$

and because the highest power of $|x_i|$ in the expansion is exactly the same as that of the binomial expansion of function $p_{\epsilon_{j-1}}(|x_i|)$, we conclude that functions $p_{\epsilon_{j-1}}(|x_i|)$ and $p_{\epsilon_j}(|x_i|)$ exhibit the same end behaviour, i.e.,

$$\lim_{|x_i|\to\infty} \frac{p_{\epsilon_{j-1}}(|x_i|)}{p_{\epsilon_j}(|x_i|)} = 1, \qquad \text{for all } j \in \mathcal{I}_q$$

(see Exercise 50, p. 139 of [100]). Therefore, we have

$$\lim_{|x_i|\to\infty} \left[p_{\epsilon_{j-1}}(|x_i|) - p_{\epsilon_j}(|x_i|)\right] = 0, \qquad \text{for all } j \in \mathcal{I}_q \tag{4.30}$$

and

$$\left[p_{\epsilon_{j-1}}(|x_i|) - p_{\epsilon_j}(|x_i|)\right]\Big|_{|x_i|=0} = \epsilon_{j-1}^p - \epsilon_j^p, \qquad \text{for all } j \in \mathcal{I}_q \tag{4.31}$$

Because function $p_{\epsilon_j}(|x_i|)$ is nondecreasing for increasing values of $|x_i|$, we obtain by combining (4.30) and (4.31)

$$0 \leq p_{\epsilon_{j-1}}(|x_i|) - p_{\epsilon_j}(|x_i|) \leq \epsilon_{j-1}^p - \epsilon_j^p, \qquad \text{for all } x_i \in \mathbb{R} \tag{4.32}$$

Furthermore, from (1.2) and (4.32), we have

$$0 \leq P_{\epsilon_{j-1}}(\boldsymbol{x}) - P_{\epsilon_j}(\boldsymbol{x}) \leq n\left(\epsilon_{j-1}^p - \epsilon_j^p\right), \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^n \tag{4.33}$$

Now the binomial expansion of $(\epsilon_{j-1}-\epsilon_j)^p$ can be written as

$$(\epsilon_{j-1}-\epsilon_j)^p = \epsilon_{j-1}^p - p\epsilon_{j-1}^{p-1}\epsilon_j + \frac{p(p-1)}{2!}\epsilon_{j-1}^{p-2}\epsilon_j^2 - \frac{p(p-1)(p-2)}{3!}\epsilon_{j-1}^{p-3}\epsilon_j^3 + \cdots, \quad \text{for all } j \in \mathcal{I}_q$$

Multiplying the negative terms by $(\epsilon_j^{p-1})/(\epsilon_j^{p-1})$ and simplifying, we can write the expansion as

$$(\epsilon_{j-1}-\epsilon_j)^p = \epsilon_{j-1}^p - \epsilon_j^p\left[p\left(\frac{\epsilon_j}{\epsilon_{j-1}}\right)^{1-p} - \frac{p(p-1)}{2!}\left(\frac{\epsilon_j}{\epsilon_{j-1}}\right)^{2-p} + \frac{p(p-1)(p-2)}{3!}\left(\frac{\epsilon_j}{\epsilon_{j-1}}\right)^{3-p} + \cdots\right] \tag{4.34}$$

and by rearranging terms, the above infinite series can be written as the power series

$$\sum_{k=1}^{\infty} a_k \left(\frac{\epsilon_j}{\epsilon_{j-1}}\right)^k \tag{4.35}$$

where

$$a_k = \frac{(-1)^{k-1}}{k!}\left(\frac{\epsilon_{j-1}}{\epsilon_j}\right)^p \prod_{l=1}^{k}(p-l+1)$$

On the basis of Definition 4.3, we have $\epsilon_j/\epsilon_{j-1} < 1$. Therefore, the series in (4.35) is convergent (see p. 775 of [100]) and can be shown to be lower bounded by

$$\sum_{k=1}^{\infty} a_k \left(\frac{\epsilon_j}{\epsilon_{j-1}}\right)^k \geq 1 \qquad \text{for all } j \in \mathcal{I}_q \tag{4.36}$$

Combining (4.34) to (4.36), we have

$$(\epsilon_{j-1} - \epsilon_j)^p \geq \epsilon_{j-1}^p - \epsilon_j^p \tag{4.37}$$

and by combining (4.33) and (4.37), we obtain (4.27) (see Theorem 1 of [48]).

Now from Corollary 4.1, function $P_{\epsilon_j}(\boldsymbol{x})$ is Lipschitz continuous with constant

$$\kappa_{\epsilon_j} = n\frac{p}{\epsilon_j^{1-p}} \tag{4.38}$$

and because Lipschitz continuity implies uniform continuity, for all $\tau > 0$ we have

$$||\boldsymbol{x} - \tilde{\boldsymbol{x}}|| < \tau/\kappa_{\epsilon_j} \quad \Rightarrow \quad |P_{\epsilon_j}(\boldsymbol{x}) - P_{\epsilon_j}(\tilde{\boldsymbol{x}})| < \tau, \qquad \text{for all } \boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathbb{R}^n \tag{4.39}$$

Furthermore, from (4.27) and (4.28), we have

$$0 \leq P_{\epsilon_{j-1}}(\boldsymbol{x}^*_{\epsilon_{j-1},\delta}) - P_{\epsilon_j}(\boldsymbol{x}^*_{\epsilon_j,\delta}) \approx P_{\epsilon_j}(\boldsymbol{x}^*_{\epsilon_{j-1},\delta}) - P_{\epsilon_j}(\boldsymbol{x}^*_{\epsilon_j,\delta}) \leq n\nu^p, \qquad \text{for all } j \in \mathcal{I}_q \tag{4.40}$$

Combining (4.38) to (4.40), we obtain (4.29), which completes the proof. □

On the basis of Lemma 4.1 and Definition 4.3, the Euclidean distance between the solutions of two consecutive problems in the sequence $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$ approaches 0 as the index j approaches q. As for the continuation procedures (see embedding algorithm in [3]), we employ an initialization strategy where the solution obtained for problem $(\mathrm{LP}_{\epsilon_{j-1},\delta})$ is used as the initial point for the solution of problem $(\mathrm{LP}_{\epsilon_j,\delta})$ for all $j \in \mathcal{I}_q$. Efficient computation of each problem is facilitated because the convergence rate of FOSs, such as that in (1.14), is improved when an appropriate initialization is employed. The Lipschitz constant of function $P_{\epsilon_j}(\boldsymbol{x})$ increases as the index j increases in the problem sequence $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$. Slow convergence can be circumvented by using an initial point that is increasingly closer to the solution of the problem at hand.

4.2.4 Moreau Envelope and Subdifferential Mapping

Let us define a function closely related to the objective function $F_{\epsilon_j,\delta}(\boldsymbol{x})$ given by

$$\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x}) = F_{\epsilon_j,\delta}(\boldsymbol{x}) - \begin{cases} n\epsilon_j^p, & \text{for } j \in \{0\} \\ P_{\epsilon_{j-1}}(\boldsymbol{x}^*_{\epsilon_{j-1},\delta}), & \text{for } j \in \mathcal{I}_q \end{cases} \tag{4.41}$$

On the basis of Definition 4.1, the problem of minimizing function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ has exactly the same solution set as the problem of minimizing function $F_{\epsilon_j,\delta}(\boldsymbol{x})$. Thus, we use both functions interchangeably hereafter. For each value of the regularization parameter ϵ_j in Definition 4.3 and for each point in the solution set $\mathcal{S}_{\epsilon_j,\delta}$ in Definition 4.1, function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ has the property specified in (1.24). This result is presented in terms of the following proposition.

Proposition 4.3 (Prox-Regularity Property at Solution Set)**.** *Let* $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ *be a sequence as described in Definition 4.3 and consider a point* $\boldsymbol{x}^*_{\epsilon,\delta}$ *in the solution set* $\mathcal{S}_{\epsilon,\delta}$*. Function* $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ *in (4.41) is subdifferentially continuous and prox-regular at* $\bar{\boldsymbol{x}} = \boldsymbol{x}^*_{\epsilon_j,\delta}$ *for* $\bar{\boldsymbol{v}} = \boldsymbol{0}$ *where* $\bar{\boldsymbol{v}} \in \boldsymbol{\partial}\widetilde{F}_{\epsilon_j,\delta}(\bar{\boldsymbol{x}})$*. Furthermore, for any* $\boldsymbol{x}^*_{\epsilon_{j-1},\delta} \in \mathcal{S}_{\epsilon_{j-1},\delta}$ *and* $\boldsymbol{x}^*_{\epsilon_j,\delta} \in \mathcal{S}_{\epsilon_j,\delta}$*, assume that* ν *is small enough in (4.27) so that the approximation*

$$P_{\epsilon_j}(\boldsymbol{x}^*_{\epsilon_j,\delta}) \approx P_{\epsilon_{j-1}}(\boldsymbol{x}^*_{\epsilon_{j-1},\delta}), \qquad \text{for all } j \in \mathcal{I}_q \tag{4.42}$$

holds true. Under this assumption, we have

$$\widetilde{F}_{\epsilon_j,\delta}(\bar{\boldsymbol{x}}) \approx 0, \qquad \text{for all } j \in \{0\}\cup\mathcal{I}_q \tag{4.43}$$

and there exists a closed ball $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ *in (4.24) so that the inequality*

$$\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x}) > -\frac{r_{\epsilon_j}}{2}||\boldsymbol{x}-\bar{\boldsymbol{x}}||^2, \qquad \text{for all } \boldsymbol{x} \neq \bar{\boldsymbol{x}} \tag{4.44}$$

holds true, where r_{ϵ_j} *is the prox-regularity parameter of* $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ *given by*

$$r_{\epsilon_j} = \frac{2\kappa_{\epsilon_j}}{\alpha} \tag{4.45}$$

and κ_{ϵ_j} *is the Lipschitz constant of* $P_{\epsilon_j}(\boldsymbol{x})$*.*

Proof. The feasible set $\mathcal{K}_\delta$ in (4.2) is closed and, based on Corollary 4.1, function $P_{\epsilon_j}(\boldsymbol{x})$ is continuous for all $j \in \{0\}\cup\mathcal{I}_q$. Thus, function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is subdifferentially continuous at $\bar{\boldsymbol{x}}$ (see Exercise 13.29 of [96]). In addition, from Example 1.11 of [96], the level sets of $F_{\epsilon_j,\delta}(\boldsymbol{x})$ given by

$$\mathcal{K}_\delta \cap \left\{\boldsymbol{x}\in\mathbb{R}^n : P_{\epsilon_j}(\boldsymbol{x}) \leq \vartheta\right\}, \qquad \text{for } \vartheta < \infty$$

are closed which implies that $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is locally lower-semicontinuous at $\bar{\boldsymbol{x}}$ (see Definition 1.1 [89]). From Theorem 3.2 of [89], function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is prox-regular at $\bar{\boldsymbol{x}}$ for $\bar{\boldsymbol{v}}$ where $\bar{\boldsymbol{v}} \in \boldsymbol{\partial}\widetilde{F}_{\epsilon_j,\delta}(\bar{\boldsymbol{x}})$.

Now by combining (4.41) with Property 1 in Definition 4.3 and (4.42), we obtain (4.43). For $j \in \{0\}\cup\mathcal{I}_q$ and $\boldsymbol{x} \notin \mathcal{K}_\delta$ or in the case where $j \in \{0\}$ and $\boldsymbol{x} \in \mathcal{K}_\delta$, the inequality in (4.44) holds true for any $r_{\epsilon_j} > 0$. For $j \in \mathcal{I}_q$ and $\boldsymbol{x} \in \mathcal{K}_\delta$, we can write (4.44) as

$$P_{\epsilon_j}(\boldsymbol{x}) - P_{\epsilon_j}(\bar{\boldsymbol{x}}) > -\frac{r_{\epsilon_j}}{2}||\boldsymbol{x}-\bar{\boldsymbol{x}}||^2, \qquad \text{for all } \boldsymbol{x} \neq \bar{\boldsymbol{x}} \tag{4.46}$$

by using (4.42). From Definition 4.1, we conclude that the left-hand side of (4.46) is nonnegative for any $\boldsymbol{x} \in B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ and that (4.44) holds true for any $r_{\epsilon_j} > 0$. If $\boldsymbol{x} \notin B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ and $P_{\epsilon_j}(\boldsymbol{x}) \leq P_{\epsilon_j}(\bar{\boldsymbol{x}})$, we can write (4.44) as

$$\kappa_{\epsilon_j}||\boldsymbol{x} - \bar{\boldsymbol{x}}|| < \frac{r_{\epsilon_j}}{2}||\boldsymbol{x} - \bar{\boldsymbol{x}}||^2, \qquad \text{for all } \boldsymbol{x} \neq \bar{\boldsymbol{x}} \tag{4.47}$$

by using Corollary 4.1 and (1.16) where κ_{ϵ_j} is the Lipschitz constant of $P_{\epsilon_j}(\boldsymbol{x})$. From (4.24), we conclude that (4.47) holds true for any r_{ϵ_j} given by (4.45). Furthermore, based on item (c) in Theorem 3.2 of [89], r_{ϵ_j} must be the prox-regularity parameter of $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$, which completes the proof. □

Prox-regular functions are nonconvex functions that exhibit unusually rich properties from an optimization perspective. As stated in the following lemma, the ME of function $\widetilde{F}_{\epsilon,\delta}(\boldsymbol{x})$ is differentiable and convex when restricted to a neighborhood of the solution of set of problem $(\mathrm{LP}_{\epsilon,\delta})$.

Lemma 4.2 (Differentiability and Convexity of Moreau Envelope)**.** *Let* $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ *be a sequence as described in Definition 4.3 and assume that the approximation in* (4.42) *holds true. Consider the ME of function* $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ *given by*

$$\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}) = \underset{\tilde{\boldsymbol{x}}}{\text{minimize}} \left[\widetilde{F}_{\epsilon_j,\delta}(\tilde{\boldsymbol{x}}) + \frac{1}{2\gamma_k}||\tilde{\boldsymbol{x}} - \boldsymbol{x}||^2\right] \tag{4.48}$$

and let $\gamma_k \in (0, 1/r_{\epsilon_j})$ *where* r_{ϵ_j} *is the prox-regularity parameter of* $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ *in* (4.45)*. There exists a closed ball* $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ *in* (4.24) *for all* $j \in \{0\} \cup \mathcal{I}_q$ *such that* $\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x})$ *with* $\boldsymbol{x} \in B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ *is (1) convex and (2) differentiable with a Lipschitz continuous gradient given by*

$$\boldsymbol{\nabla}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}) = \frac{1}{\gamma_k}\left\{\boldsymbol{x} - \mathrm{prox}_{\gamma_k}\left[\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})\right]\right\} \tag{4.49}$$

Proof. For all $j \in \{0\} \cup \mathcal{I}_q$, function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is subdifferentially continuous and prox-regular at $\bar{\boldsymbol{x}} = \boldsymbol{x}^*_{\epsilon_j,\delta}$ for $\bar{\boldsymbol{v}} = \boldsymbol{0}$ where $\bar{\boldsymbol{v}} \in \boldsymbol{\partial}\widetilde{F}_{\epsilon_j,\delta}(\bar{\boldsymbol{x}})$ (see Proposition 4.3). Hence, results in [89] concerning the operator $\mathrm{T}_{\epsilon_j,\delta}$ as an $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$-attentive localization of $\boldsymbol{\partial}\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ can be restated in terms of ordinary localization (see Remark 5.9 of [89]). In addition, we conclude from (4.43) and (4.44) that Assumption (4.1) of [89] holds true for $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ at the values of $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{v}}$ under consideration. Thus, for $\gamma_k \in (0, 1/r_{\epsilon_j})$ with r_{ϵ_j} given by (4.45), there exists a closed ball $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ in (4.24) such that

function $\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x})$ with $\boldsymbol{x} \in B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ is (1) convex (see item (a) in Proposition 5.4 of [89]) and (2) differentiable with a Lipschitz continuous gradient given by (4.49) (see Theorem 4.4 of [89]), which completes the proof. □

On the basis of Lemma 4.2, each problem in the sequence $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$ can be solved efficiently because the update formula in (1.20) with β_k given by (1.27) is applicable (see Sec. 3.2.1 of [90] and Theorem 1 of [81]).

Another property of interest in prox-regular functions is related to their subdifferential mapping. The operator $\mathrm{T}_{\epsilon,\delta}(\boldsymbol{x}) \triangleq \partial\widetilde{F}_{\epsilon,\delta}(\boldsymbol{x})$ is said to be monotone if it possesses the property

$$(\boldsymbol{g}_1 - \boldsymbol{g}_0)^T(\boldsymbol{x}_1 - \boldsymbol{x}_0) \geq 0$$

whenever $\boldsymbol{g}_0 \in \mathrm{T}_{\epsilon,\delta}(\boldsymbol{x}_0)$ and $\boldsymbol{g}_1 \in \mathrm{T}_{\epsilon,\delta}(\boldsymbol{x}_1)$ (see Definition 12.1 of [96]). Because the feasible set $\mathcal{K}_\delta$ in (4.2) is convex, it follows that the indicator function $I_{\mathcal{K}_\delta}(\boldsymbol{x})$ in (4.5) is also convex (see p. 28 of [95]). Therefore, the operator $\mathrm{T}_{\epsilon,\delta}$ is monotone when $P_\epsilon(\boldsymbol{x})$ is a convex function since $\widetilde{F}_{\epsilon,\delta}(\boldsymbol{x})$ reduces to the sum of two convex functions which has a monotone subdifferential mapping (see Theorem 12.17 of [96]). The sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.14) has been shown to converge to a solution to the inclusion

$$\mathrm{T}_{\epsilon,\delta}(\boldsymbol{x}) \ni \mathbf{0} \tag{4.50}$$

for any $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ and $\gamma_k > 0$ when $\mathrm{T}_{\epsilon,\delta}$ is monotone (see Theorem 1 of [94]). Thus, PP methods are applicable to the solution of convex recovery problems. Unfortunately, monotonicity assumptions are not valid for the recovery problem at hand because $\widetilde{F}_{\epsilon,\delta}(\boldsymbol{x})$ is a nonconvex function as $P_\epsilon(\boldsymbol{x})$ is based on an SPF of class $\mathcal{N}$.

The convergence of PP methods under relaxed monotonicity of the subdifferential mapping (or of its inverse) has been addressed in [30, 63, 87]. Sequence $\left\{\boldsymbol{x}^{(k)}\right\}_{k\in\mathbb{N}}$ in (1.14) has been shown to converge to a solution to the inclusion in (4.50) provided that $\boldsymbol{x}^{(0)}$ is close enough to the solution set and γ_k is large enough (see Theorem 3.1 of [30]). Relaxed monotonicity of the subdifferential mapping in these convergence conditions is based on the concept of cohypomonotonicity. The operator $\mathrm{T}_{\epsilon,\delta}$ is said to be η_ϵ-cohypomonotone if there exists a constant $\eta_\epsilon > 0$ such that the mapping

$$\mathrm{T}_{\epsilon,\delta}^{-1} + \eta_\epsilon \,\mathrm{Id}$$

is monotone where Id denotes the identity mapping (see Definition 2.2 of [30]). The relaxed monotonicity of the subdifferential mapping of prox-regular functions has been

identified in [89]. On the basis of Proposition 4.3 and Proposition 5.4 of [89], the following lemma states that for each value of the regularization sequence ϵ_j in Definition 4.3, the subdifferential mapping of $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is cohypomonotone when restricted to a neighborhood of the solution set $\mathcal{S}_{\epsilon_j,\delta}$.

Lemma 4.3 (Cohypomonotonicity of Subdifferential)**.** *Let $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ be a sequence as described in Definition 4.3 and assume that the approximation in (4.42) holds true. There exists a closed ball $B(\alpha,\mathcal{S}_{\epsilon_j,\delta})$ in (4.24) for all $j\in\{0\}\cup\mathcal{I}_q$ such that the operator $\mathrm{T}_{\epsilon_j,\delta}$ associated with function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is η_{ϵ_j}-cohypomonotone on $B(\alpha,\mathcal{S}_{\epsilon_j,\delta})$ for*

$$\eta_{\epsilon_j}\in(0,1/r_{\epsilon_j}) \tag{4.51}$$

where r_{ϵ_j} is the prox-regularity parameter of $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ in (4.45).

Proof. For all $j\in\{0\}\cup\mathcal{I}_q$, function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is subdifferentially continuous and prox-regular at $\bar{\boldsymbol{x}}=\boldsymbol{x}^*_{\epsilon_j,\delta}$ for $\bar{\boldsymbol{v}}=\boldsymbol{0}$ where $\bar{\boldsymbol{v}}\in\boldsymbol{\partial}\widetilde{F}_{\epsilon_j,\delta}(\bar{\boldsymbol{x}})$ (see Proposition 4.3). Hence, due to these properties, results in [89] concerning the operator $\mathrm{T}_{\epsilon_j,\delta}$ as an $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$-attentive localization of $\boldsymbol{\partial}\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ can be restated in terms of ordinary localization (see Remark 6.9 of [89]). In addition, we conclude from (4.43) and (4.44) that Assumption (4.1) of [89] holds true for $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ at the values of $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{v}}$ under consideration. Thus, there exists a closed ball $B(\alpha,\mathcal{S}_{\epsilon_j,\delta})$ in (4.24) such that the operator $\mathrm{T}_{\epsilon_j,\delta}$ associated with function $\widetilde{F}_{\epsilon_j,\delta}(\boldsymbol{x})$ is η_{ϵ_j}-cohypomonotone on $B(\alpha,\mathcal{S}_{\epsilon_j,\delta})$ for η_{ϵ_j} given by (4.51) (see item (b) in Proposition 5.4 of [89]), which completes the proof. □

On the basis of Lemma 4.3, each problem in the sequence $\big\{(\mathrm{LP}_{\epsilon_j,\delta})\big\}_{j\in\{0\}\cup\mathcal{I}_q}$ can be solved by using PP methods when the convergence conditions specified in Theorem 3.1 of [30] are satisfied.

4.3 Inexact PP Based BP Method

In this section, we propose a new PP method for the solution of the recovery problem in (4.3). Suppose that for all $j\in\{0\}\cup\mathcal{I}_q$, we can find an initial point $\boldsymbol{x}^{(0)}_{\epsilon_j}$ within a closed ball $B(\alpha,\mathcal{S}_{\epsilon_j,\delta})$ specified in (4.24). By combining (1.14), (4.3), and (4.49), we

obtain

$$\begin{aligned}\boldsymbol{x}_{\epsilon_j}^{(k+1)} &= \boldsymbol{x}_{\epsilon_j}^{(k)} - \gamma_k \boldsymbol{\nabla}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)}) \\ &= \arg \underset{\boldsymbol{x}}{\text{minimize}} \left[P_{\epsilon_j}(\boldsymbol{x}) + I_{\mathcal{K}_\delta}(\boldsymbol{x}) + \frac{1}{2\gamma_k}||\boldsymbol{x}-\boldsymbol{x}_{\epsilon_j}^{(k)}||^2 \right], \quad \text{for all } j \in \{0\}\cup\mathcal{I}_q\end{aligned} \tag{4.52}$$

where $\gamma_k \in (0, 1/r_{\epsilon_j})$ for all $k \in \mathbb{N}$. On the basis of Lemma 4.2, the ME of function $F_{\epsilon_j,\delta}(\boldsymbol{x})$ is differentiable and, consequently, the above update formula is applicable to the solution of each problem in the sequence $\left\{(\text{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$. Specifically, each solution point $\boldsymbol{x}_{\epsilon_j}^{(k)}$ lies within the same closed ball $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ as the sequence $\left\{\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}^{(k)})\right\}_{k\in\mathbb{N}}$ associated with the update formula in (4.52) is monotonically decreasing (see Sec. 1.2.1 of [82]). When the minimization problem in (4.48) is solved exactly, the above update formula is said to be in *exact form*.

As stated in the following proposition, the proposed PP method is closely related to projected subgradient methods when in exact form.

Proposition 4.4 (Update Formula in Exact Form)**.** *Consider the update formula in* (4.52) *and assume that* $\boldsymbol{x}_{\epsilon_j}^{(k)} \in B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ *for all* $k \in \mathbb{N}$ *and* $j \in \{0\} \cup \mathcal{I}_q$*. Under this assumption, the update formula can be written as*

$$\boldsymbol{x}_{\epsilon_j}^{(k+1)} = \text{proj}_{\mathcal{K}_\delta}\left[\boldsymbol{x}_{\epsilon_j}^{(k)} - \gamma_k\, \boldsymbol{g}_{\epsilon_j}^{(k+1)}\right], \qquad \textit{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.53}$$

where

$$\boldsymbol{g}_{\epsilon_j}^{(k+1)} \in \boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k+1)}) \tag{4.54}$$

Proof. The differentiability and convexity of function $\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x})$ imply that the subproblem in (4.52) is convex because it can be written as the minimization of the sum of the linear approximating function of $\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x})$ and a convex penalty function (see pp. 21 of [90]). Hence, vector $\boldsymbol{x}_{\epsilon_j}^{(k+1)}$ is a minimizer if and only if the zero vector belongs to the subdifferential of the sum of the functions $P_{\epsilon_j}(\boldsymbol{x})$, $I_{\mathcal{K}_\delta}(\boldsymbol{x})$, and $1/(2\gamma_k)||\boldsymbol{x}-\boldsymbol{x}_{\epsilon_j}^{(k)}||^2$ and (4.52) holds if and only if

$$\begin{aligned}&\partial\left[P_{\epsilon_j}(\cdot) + I_{\mathcal{K}_\delta}(\cdot) + \frac{1}{2\gamma_k}||(\cdot) - \boldsymbol{x}_{\epsilon_j}^{(k)}||^2\right](\boldsymbol{x}_{\epsilon_j}^{(k+1)}) \ni \mathbf{0} \\ &\partial\left[P_{\epsilon_j}(\cdot) + \frac{1}{2\gamma_k}||(\cdot) - \boldsymbol{x}_{\epsilon_j}^{(k)}||^2\right](\boldsymbol{x}_{\epsilon_j}^{(k+1)}) + N_{\mathcal{K}_\delta}(\boldsymbol{x}_{\epsilon_j}^{(k+1)}) \ni \mathbf{0}\end{aligned}, \quad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.55}$$

(see Theorem 8.15 of [96]). We write (4.55) as

$$\frac{1}{\gamma_k}(\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)}) \in \partial P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k+1)}) + N_{\mathcal{K}_\delta}(\boldsymbol{x}_{\epsilon_j}^{(k+1)}), \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.56}$$

and (4.56) holds true if and only if

$$\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)} - \gamma_k\, \boldsymbol{g}_{\epsilon_j}^{(k+1)} \in N_{\mathcal{K}_\delta}(\boldsymbol{x}_{\epsilon_j}^{(k+1)}), \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.57}$$

for some $\boldsymbol{g}_{\epsilon_j}^{(k+1)} \in \boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k+1)})$. From (4.22), the inclusion in (4.57) is equivalent to

$$\left[\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)} - \gamma_k\, \boldsymbol{g}_{\epsilon_j}^{(k+1)}\right]^T (\boldsymbol{x} - \boldsymbol{x}_{\epsilon_j}^{(k+1)}) \leq 0, \quad \forall\, \boldsymbol{x} \in \mathcal{K}_\delta$$

and since $\boldsymbol{x}_{\epsilon_j}^{(k)} \in \mathcal{K}_\delta$, we obtain

$$\left[\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)} - \gamma_k\, \boldsymbol{g}_{\epsilon_j}^{(k+1)}\right]^T (\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)}) \leq 0 \tag{4.58}$$

which is true if and only if (4.53) holds by the projection theorem (see Proposition 1.1.9 of [12]), which completes the proof. □

On the basis of Proposition 4.4, the update formula in (4.53) is similar to that in (1.10) with the only difference being that the subgradient is evaluated at a different point. The update formula in (4.53) is preferred over that in (4.52) because the projector onto set $\mathcal{K}_\delta$ can be computed efficiently. Unfortunately, (4.54) is of little practical use because the subgradient is evaluated at the next solution point $\boldsymbol{x}_{\epsilon_j}^{(k+1)}$ rather than at the current solution point $\boldsymbol{x}_{\epsilon_j}^{(k)}$.

In Sec. 4.3.1, we show that for approximate solutions to the minimization problem in (4.48), the resulting inexact update formula is applied by iteratively performing two fundamental operations. In Sec. 4.3.2, we show that the first operation can be performed either analytically or numerically as the limit of an infinite series of nested radicals. In Sec. 4.3.3, we show that the second operation can be performed by using a fast convergent version of the AP method. In Sec. 4.3.4, we present convergence results for the proposed method. In Sec. 4.3.5, we propose a two-step method for accelerated convergence.

4.3.1 Proposed FOS

An inexact variant of the update formula in (4.53) is given by

$$\boldsymbol{x}_{\epsilon_j}^{(k+1)} = \text{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{x}_{\epsilon_j}^{(k)} - \gamma_k\, \tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)}\right), \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.59}$$

where

$$\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)} \in \boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k+1)} + \boldsymbol{e}_{\epsilon_j}^{(k)}) \tag{4.60}$$

is a subgradient of function $P_{\epsilon_j}(\boldsymbol{x})$, column vector $\boldsymbol{e}_{\epsilon_j}^{(k)}$ given by

$$\boldsymbol{e}_{\epsilon_j}^{(k)} = \boldsymbol{z}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)} \tag{4.61}$$

is the error introduced in (4.53), and $\boldsymbol{z}_{\epsilon_j}^{(k)}$ is a solution to the inclusion

$$\frac{1}{\gamma_k}\left(\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k)}\right) \in \boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{z}_{\epsilon_j}^{(k)}), \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.62}$$

Use of the update formula in (4.59) approximately solves the minimization problem in (4.48). The approximation error is proportional to the magnitude of vector $\boldsymbol{e}_{\epsilon_j}^{(k)}$ in (4.61). Because the subgradient is evaluated at point $\boldsymbol{z}_{\epsilon_j}^{(k)}$, (4.60) is applicable unlike (4.54) where the subgradient is evaluated at the next solution point $\boldsymbol{x}_{\epsilon_j}^{(k+1)}$.

As stated in the following proposition, the proposed update formula is applied by iteratively performing two fundamental operations.

Proposition 4.5 (Inexact Variant of the Update Formula)**.** *The update formula in* (4.59) *can be written as the iterative computation of the PP of function* $P_{\epsilon_j}(\boldsymbol{x})$ *followed by the projection of such a point into the feasible set* $\mathcal{K}_\delta$*, namely,*

$$\boldsymbol{x}_{\epsilon_j}^{(k+1)} = \text{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{z}_{\epsilon_j}^{(k)}\right), \qquad \textit{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.63}$$

where

$$\boldsymbol{z}_{\epsilon_j}^{(k)} \in \text{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})\right] \tag{4.64}$$

Proof. The inclusion in (4.62) holds true if and only if

$$\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)} = \frac{1}{\gamma_k}\left(\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k)}\right), \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.65}$$

and for some $\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)} \in \boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{z}_{\epsilon_j}^{(k)})$. Combining (4.59) and (4.65), we obtain (4.63). From

Corollary 4.1, $P_{\epsilon_j}(\boldsymbol{x})$ is a lower-C^2 function and, therefore, $\boldsymbol{z}_{\epsilon_j}^{(k)}$ is a PP of $P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})$ because (4.65) implies (4.64) (see Theorem 1 of [56]), which completes the proof. □

The proposed update formula in (4.63) defines what we call the iterative proximal-point projection (IPPP) method. If we assume that the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ associated with (4.63) converges to a solution point $\boldsymbol{x}_{\epsilon_j}^* \in \mathcal{S}_{\epsilon_j,\delta}$ for all $j \in \{0\} \cup \mathcal{I}_q$, then the convergence of such a sequence can be quite slow as index j approaches q because function $P_{\epsilon_j}(\boldsymbol{x})$ may assume arbitrarily large Lipschitz constants (see Sec. 4.2.3 for details). This problem is circumvented by employing initialization strategies such as those used in continuation procedures (see embedding algorithm in [3]). Continuation procedures are applicable to the problem at hand because function $P_{\epsilon_j}(\boldsymbol{x})$ with $j \in \{0\}\cup\mathcal{I}_q$ defines a homotopy (or deformation) of the ℓ_p^p norm of $\boldsymbol{x}$ as the regularization sequence $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ is a strictly decreasing sequence where the approximations

$$P_{\epsilon_0}(\boldsymbol{x}) \approx n\epsilon_0^p \qquad \text{and} \qquad P_{\epsilon_q}(\boldsymbol{x}) \approx ||\boldsymbol{x}||_p^p$$

hold true (see Definition 4.3 and Chapter 1 of [3]).

For the initial index $j = 0$, we choose an initial point $\boldsymbol{x}_{\epsilon_0}^{(0)}$ and let the initial point for the remaining indices be given by

$$\boldsymbol{x}_{\epsilon_j}^{(0)} = \boldsymbol{x}_{\epsilon_{j-1}}^*, \qquad \text{for all } j \in \mathcal{I}_q \tag{4.66}$$

On the basis of Lemma 4.1, the convergence rate of the IPPP method is improved by using such an initialization strategy. The Euclidean distance between solutions of two consecutive problems in the sequence $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$ approaches 0 as the index j approaches q. Consequently, initial points employed are increasingly closer to the solution of the problem at hand for increasing j. The IPPP method with continuation is defined in Algorithm 4.1.

4.3.2 Computation of the PP

Here we are searching for an efficient method for computing the PP of function $P_{\epsilon_j}(\boldsymbol{x})$. From (1.2) and (1.15), the PP mapping of $P_{\epsilon_j}(\boldsymbol{x})$ is defined by

$$\mathrm{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x})\right] = \arg\ \underset{\bar{\boldsymbol{x}}}{\text{minimize}}\ \left[\sum_{i=1}^{n} p_{\epsilon_j}(|x_i|) + \frac{1}{2\gamma_k}||\bar{\boldsymbol{x}} - \boldsymbol{x}||^2\right],\ \text{for all } j \in \{0\}\cup\mathcal{I}_q \tag{4.67}$$

Input: $\boldsymbol{x}^{(0)} \in \mathcal{K}_\delta$, $\varepsilon_c > 0$, $\{\epsilon_j\}_{j\in\mathcal{I}_q}$, $\gamma_k \in (0, 1/r_{\epsilon_j})$
Output: $\boldsymbol{x}^*_{\epsilon_q} \in \mathcal{S}_{\epsilon_q,\delta}$
$\boldsymbol{x}^*_{\epsilon_{-1}} = \boldsymbol{x}^{(0)}$;
for $j = 0$ **to** q **do**
  $\boldsymbol{x}^{(0)}_{\epsilon_j} = \boldsymbol{x}^*_{\epsilon_{j-1}}$;
  $k = -1$;
  repeat
    $k = k + 1$;
    Compute the PP of $P_{\epsilon_j}(\boldsymbol{x}^{(k)}_{\epsilon_j})$ (see Algorithm 4.2);
    $\boldsymbol{z}^{(k)}_{\epsilon_j} = \text{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x}^{(k)}_{\epsilon_j})\right]$;
    Compute the projection of $\boldsymbol{z}^{(k)}_{\epsilon_j}$ onto set $\mathcal{K}_\delta$ (see Algorithm 4.3);
    $\boldsymbol{x}^{(k+1)}_{\epsilon_j} = \text{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{z}^{(k)}_{\epsilon_j}\right)$;
  until $||F_{\epsilon,\delta}(\boldsymbol{x}^{(k+1)}_{\epsilon_j}) - F_{\epsilon,\delta}(\boldsymbol{x}^{(k)}_{\epsilon_j})|| \leq \varepsilon_c$;
  $\boldsymbol{x}^*_{\epsilon_j} = \boldsymbol{x}^{(k+1)}_{\epsilon_j}$;
end

Algorithm 4.1: IPPP Method with Continuation

and is closely related to the PP mapping of function $p_{\epsilon_j}(|x_i|)$ which is defined as

$$\text{prox}_{\gamma_k}\left[p_{\epsilon_j}(|x_i|)\right] = \arg\ \underset{\bar{x}_i}{\text{minimize}}\left[p_{\epsilon_j}(|x_i|) + \frac{1}{2\gamma_k}|\bar{x}_i - x_i|^2\right],\ \text{for all } j \in \{0\}\cup\mathcal{I}_q \tag{4.68}$$

The optimization problems in (4.67) and (4.68) have an unique solution for appropriate values of the prox-parameter γ_k and the solution of the problem in (4.67) can be obtained from that of the problem in (4.68). These results are stated in the following proposition.

Proposition 4.6 (Uniqueness of PP)**.** *Let $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ be a sequence as described in Definition 4.3 and let ρ_{ϵ_j} denote the lower-C^2 constant of $P_{\epsilon_j}(\boldsymbol{x})$. When $\gamma_k < 1/\rho_{\epsilon_j}$, the PPs $\boldsymbol{z}$ and z_i are uniquely determined by the relations*

$$\boldsymbol{z} = \text{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x})\right] \iff \frac{1}{\gamma_k}(\boldsymbol{x}-\boldsymbol{z}) \in \boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{z}), \quad \textit{for all } j\in\{0\}\cup\mathcal{I}_q \tag{4.69a}$$

$$z_i = \text{prox}_{\gamma_k}\left[p_{\epsilon_j}(|x_i|)\right] \iff \frac{1}{\gamma_k}(x_i-z_i) \in \partial\, p_{\epsilon_j}(|z_i|), \ \textit{for all } j\in\{0\}\cup\mathcal{I}_q \tag{4.69b}$$

The PP mapping of $P_{\epsilon_j}(\boldsymbol{x})$ *is related to that of* $p_{\epsilon_j}(|x_i|)$ *by*

$$\operatorname{prox}_{\gamma_k}\{P_{\epsilon_j}(\boldsymbol{x})\} = \begin{bmatrix}\operatorname{prox}_{\gamma_k}\{p_{\epsilon_j}(|x_1|)\} & \cdots & \operatorname{prox}_{\gamma_k}\{p_{\epsilon_j}(|x_n|)\}\end{bmatrix}^T \tag{4.70}$$

and the PPs $\boldsymbol{z}$ *and* z_i *are related by*

$$\boldsymbol{z} = \begin{bmatrix} z_1 & z_2 & \cdots & z_n \end{bmatrix}^T \tag{4.71}$$

Proof. For all $j \in \{0\} \cup \mathcal{I}_q$, functions $p_{\epsilon_j}(|x_i|)$ and $P_{\epsilon_j}(\boldsymbol{x})$ are bounded from below by ϵ_j and $n\epsilon_j$, respectively. The uniqueness of the PPs of functions $p_{\epsilon_j}(|x_i|)$ and $P_{\epsilon_j}(\boldsymbol{x})$ follows from their boundedness and from Proposition 4.2 and Corollary 4.1 (see Theorem 1 of [56]). Thus, we obtain 4.69a and (4.69b).

The subdifferential of $P_{\epsilon_j}(\boldsymbol{x})$ is related to that of $p_{\epsilon_j}(|x_i|)$ by

$$\boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{x}) = \prod_{i=1}^{n} \partial p_{\epsilon_j}(|x_i|), \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.72}$$

(see Corollary 2.4.5 of [116]). Using a similar approach to that in the proof of Lemma 2.9 of [31], it follows from 4.69a and (4.72) that

$$\boldsymbol{z} = \operatorname{prox}_{\gamma_k}\{P_{\epsilon_j}(\boldsymbol{x})\} \quad \Longleftrightarrow \quad \frac{1}{\gamma_k}(\boldsymbol{x} - \boldsymbol{z}) \in \boldsymbol{\partial} P_{\epsilon_j}(\boldsymbol{z}) = \prod_{i=1}^{n} \partial p_{\epsilon_j}(|z_i|) \tag{4.73}$$

which is true if and only if

$$\boldsymbol{z} = \begin{bmatrix}\operatorname{prox}_{\gamma_k}\{p_{\epsilon_j}(|x_1|)\} & \cdots & \operatorname{prox}_{\gamma_k}\{p_{\epsilon_j}(|x_n|)\}\end{bmatrix} \tag{4.74}$$

Combining 4.69, (4.73), and (4.74), we obtain (4.70) and (4.71), which completes the proof. □

On the basis of (4.11) and Proposition 4.6, we can express the PP z_i in (4.69b) as

$$z_i = \begin{cases} z_-, & x_i \in \mathbb{R}^- \\ z_+, & x_i \in \mathbb{R}^+ \\ 0, & x_i = 0 \end{cases} \tag{4.75}$$

where z_- and z_+ are PPs determined by the relations

$$\begin{aligned} z_- = \text{prox}_{\gamma_k}^{-}\left[p_{\epsilon_j,-}(x_i)\right] < 0 \quad &\Longleftrightarrow \quad p\,(-z_- + \epsilon_j)^{p-1} = \frac{1}{\gamma_k}(z_- - x_i) \\ z_+ = \text{prox}_{\gamma_k}^{+}\left[p_{\epsilon_j,+}(x_i)\right] > 0 \quad &\Longleftrightarrow \quad p\,(z_+ + \epsilon_j)^{p-1} = -\frac{1}{\gamma_k}(z_+ - x_i) \end{aligned} \tag{4.76}$$

and the PP mappings of functions $p_{\epsilon_j,-}(x_i)$ and $p_{\epsilon_j,+}(x_i)$ are defined by

$$\begin{aligned} \text{prox}_{\gamma_k}^{-}\left[p_{\epsilon_j,-}(x_i)\right] &= \arg \underset{x\in\mathbb{R}^-}{\text{minimize}}\left[p_{\epsilon_j,-}(x_i) + \frac{1}{2\gamma_k}|x - x_i|^2\right] \\ \text{prox}_{\gamma_k}^{+}\left[p_{\epsilon_j,+}(x_i)\right] &= \arg \underset{x\in\mathbb{R}^+}{\text{minimize}}\left[p_{\epsilon_j,+}(x_i) + \frac{1}{2\gamma_k}|x - x_i|^2\right] \end{aligned}, \quad \text{for all } j \in \{0\} \cup \mathcal{I}_q$$

The computation of the PP z_i is facilitated when p in (1.4) is expressed as a common fraction of the form

$$p = \frac{l-1}{l}, \qquad \text{for any } l \in \{2,\ 3,\ \ldots\} \tag{4.77}$$

and the PPs z_- and z_+ are employed. Combining (4.76) and (4.77), we obtain

$$z_- = \text{prox}_{\gamma_k}^{-}\left[p_{\epsilon_j,-}(x_i)\right] < 0 \quad \Longleftrightarrow \quad \frac{l-1}{l}(-z_- + \epsilon_j)^{-\frac{1}{l}} = \frac{1}{\gamma_k}(z_- - x_i) \tag{4.78a}$$

$$z_+ = \text{prox}_{\gamma_k}^{+}\left[p_{\epsilon_j,+}(x_i)\right] > 0 \quad \Longleftrightarrow \quad \frac{l-1}{l}(z_+ + \epsilon_j)^{-\frac{1}{l}} = -\frac{1}{\gamma_k}(z_+ - x_i) \tag{4.78b}$$

and substituting new variables v_- and v_+ given by

$$\begin{aligned} v_- &= (-z_- + \epsilon_j)^{\frac{1}{l}} \\ v_+ &= (z_+ + \epsilon_j)^{\frac{1}{l}} \end{aligned} \tag{4.79}$$

into 4.78a and (4.78b), the relations in (4.78) hold true if and only if

$$-v_-^l + \epsilon_j = \text{prox}_{\gamma_k}^{-}\left[p_{\epsilon_j,-}(x_i)\right] < 0 \quad \Longleftrightarrow \quad v_-^{l+1} + (x_i - \epsilon_j)v_- + \gamma_k\frac{l-1}{l} = 0 \tag{4.80a}$$

$$+v_+^l - \epsilon_j = \text{prox}_{\gamma_k}^{+}\left[p_{\epsilon_j,+}(x_i)\right] > 0 \quad \Longleftrightarrow \quad v_+^{l+1} + (-x_i - \epsilon_j)v_+ + \gamma_k\frac{l-1}{l} = 0 \tag{4.80b}$$

hold true. Thus, the PPs z_- and z_+ can be computed by solving trinomial equations of the form in (4.80). As stated in the following proposition, the computation of the PP z_i boils down to finding the largest root of a trinomial equation which is obtained

by combining the trinomial equations on the right-hand side of 4.80a and (4.80b).

Proposition 4.7 (PP and Trinomial Equation)**.** *Let* $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ *be a sequence as described in Definition 4.3 and assume that* $\gamma_k < 1/\rho_{\epsilon_j}$ *for all* $j \in \{0\}\cup\mathcal{I}_q$*. The PP* z_i *in* (4.69b) *is given by*

$$z_i = \begin{cases} \text{sign}(x_i)(v^l - \epsilon_j), & |x_i| > \gamma_k \dfrac{l-1}{l\sqrt[l]{\epsilon_j}} \\ 0, & \textit{otherwise} \end{cases}, \qquad \textit{for all } j \in \{0\}\cup\mathcal{I}_q \tag{4.81}$$

where v *is the largest real root of the trinomial equation*

$$v^{l+1} + (-|x_i| - \epsilon_j)v + \gamma_k \frac{l-1}{l} = 0 \tag{4.82}$$

Proof. On the basis of the uniqueness of z_i in (4.69b) and from (4.75), (4.76), and (4.78) to (4.80), we conclude that v_- and v_+ are unique real roots greater than $\sqrt[l]{\epsilon_j}$ of the trinomial equations on the right-hand side of 4.80a and (4.80b), respectively. The number of real roots greater than $\sqrt[l]{\epsilon_j}$ in the trinomial equation in (4.80a) is the same as the number of positive real roots of the transformed equation obtained from the original one by using the substitution

$$\bar{v}_- = \sqrt[l]{\epsilon_j} + v_- \tag{4.83}$$

(see p. 126 of [110]). Combining 4.80a and (4.83), the transformed equation is given by

$$\left(\sqrt[l]{\epsilon_j} + \bar{v}_-\right)^{l+1} + (x_i - \epsilon_j)\left(\sqrt[l]{\epsilon_j} + \bar{v}_-\right) + \gamma_k \frac{l-1}{l} = 0 \tag{4.84}$$

which after expanding the binomials and simplifying is equivalent to the polynomial

$$a_{l+1}\bar{v}_-^{l+1} + a_l\bar{v}_-^{l} + a_{l-1}\bar{v}_-^{l-1} + \ldots + a_2\bar{v}_-^{2} + a_1\bar{v}_- + a_0 = 0 \tag{4.85}$$

where

$$a_\ell = \begin{cases} 1, & \ell = l+1 \\ \binom{l+1}{\ell}\epsilon_j^{\frac{l+1-\ell}{l}}, & \ell \in \{l,\ l-1,\ \ldots,\ 3,\ 2\} \\ \left[\binom{l+1}{\ell} - 1\right]\epsilon_j + x_i, & \ell = 1 \\ \gamma_k \frac{l-1}{l} + \sqrt[l]{\epsilon_j}x_i, & \ell = 0 \end{cases} \tag{4.86}$$

Let $\mathcal{I}_l = \{l+1,\ l,\ l-1,\ \ldots,\ 3,\ 2\}$ and let $\mathcal{I}_{l+2} = \mathcal{I}_l \cup \{1,\ 0\}$. The number of

positive real roots of the polynomial in (4.85) is never greater than the number of sign changes in the sequence $\{a_\ell\}_{\ell\in\mathcal{I}_{l+2}}$ (see Descartes' rule of signs, p. 121 of [110]). Since the trinomial equation in (4.80a) has an unique real root greater than $\sqrt[l]{\epsilon_j}$, the polynomial equation in (4.85) must have an unique positive real root. Therefore, the sequence $\{a_\ell\}_{\ell\in\mathcal{I}_{l+2}}$ must have at least one sign change. From (4.86), we have $a_\ell > 0$ for $\ell \in \mathcal{I}_l$ while a_1 can be either positive, negative, or zero. Thus, $\{a_\ell\}_{\ell\in\mathcal{I}_{l+2}}$ has one sign change when $a_0 < 0$ or when

$$x_i < -\gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} \tag{4.87a}$$

By using a similar approach to that used for the trinomial equation in (4.80a), it can be shown that the existence of a unique real root greater than $\sqrt[l]{\epsilon_j}$ in the trinomial equation in (4.80b) implies the inequality

$$x_i > \gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} \tag{4.87b}$$

From (4.75), (4.76), (4.78), (4.80), and (4.87), we obtain (4.81) and, by combining the trinomial equations on the right-hand side of 4.80a and (4.80b), we obtain (4.82). Lastly, since v is the unique real root greater than $\sqrt[l]{\epsilon_j}$ of (4.82), it must also be the largest root, which completes the proof. □

On the basis of Proposition 4.7, the PP z_i in (4.69b) can be computed efficiently when the trinomial equation in (4.82) is solved either analytically or numerically by using a fast iterative method. In the general case where $l \in \{2,\ 3,\ \ldots\}$, a solution is obtained as the limit of nested series with infinitely many terms such as

$$cf\left\{x + cf\left[x + cf(x)\right]\right\}\cdots \tag{4.88}$$

where c is a real constant. For example, if we let $f(x) = \sqrt{x}$ in (4.88), then the resulting infinite series reduces to so-called infinite radicals (see Section II of [57]). The roots of trinomial equations can be approximated as the limit of infinite radicals, as is done in [99] or by employing the so-called Bolyai algorithm (see Section 2 of [103]). The infinite series involved in such approximations have fast convergence. For $l = 2$ or 3, (4.82) reduces to a polynomial equation of third and fourth degree, respectively. The roots of cubic and biquadratic equations can be obtained analytically (see Chapter V of [110]). On the basis of these results, the largest real root of the trinomial

equation in (4.82) is described in the following proposition.

Proposition 4.8 (Root of the Trinomial Equation)**.** *Let* $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ *be a sequence as described in Definition 4.3. Assume that* $\gamma_k < 1/\rho_{\epsilon_j}$ *for all* $j \in \{0\}\cup\mathcal{I}_q$ *and* p *is of the form in (4.77). The largest real root of the trinomial equation in (4.82) is given by the limit of infinite radicals of the form*

$$v = \sqrt[l+1]{-\gamma_k\frac{l-1}{l} - (-|x_i| - \epsilon_j)\sqrt[l+1]{-\gamma_k\frac{l-1}{l} - (-|x_i| - \epsilon_j)\sqrt[l+1]{-\gamma_k\frac{l-1}{l} - \cdots}}} \tag{4.89}$$

for all $l \in \{2,\ 3,\ \ldots\}$*. For* $l = 2$ *or* 3*, if we let*

$$c = \sqrt[3]{-\frac{b}{2} + \sqrt{\frac{b^2}{4} + \frac{a^3}{27}}} \tag{4.90}$$

where

$$b = \begin{cases} \gamma_k\frac{l-1}{l}, & l = 2 \\ -(-|x_i| - \epsilon_j)^2, & l = 3 \end{cases} \tag{4.91a}$$

$$a = \begin{cases} -|x_i| - \epsilon_j, & l = 2 \\ -4\gamma_k\frac{l-1}{l}, & l = 3 \end{cases} \tag{4.91b}$$

then the largest real root has a closed-form expression given by

$$v = \begin{cases} c - \frac{a}{3c}, & l = 2 \\ \frac{1}{2}\left[\sqrt{c - \frac{a}{3c}} + \sqrt{4\sqrt{\frac{1}{4}\left(c - \frac{a}{3c}\right)^2 - \gamma_k\frac{l-1}{l}} - \left(c - \frac{a}{3c}\right)}\right], & l = 3 \end{cases} \tag{4.92}$$

Proof. Assume the contrary, namely, that for all ϵ_j with $j \in \{0\}\cup\mathcal{I}_q$ and for all $l \in \{2,\ 3,\ \ldots\}$ the inequality given by

$$\left(\frac{\gamma_k\frac{l}{l-1}}{l}\right)^l > \left(\frac{-|x_i| - \epsilon_j}{l+1}\right)^{l+1} \tag{4.93}$$

holds true for

$$\gamma_k < 1/\rho_{\epsilon_j} \quad \Rightarrow \quad \gamma_k < \frac{l-1}{l^2\epsilon_j^{\frac{l-1}{l}}} \tag{4.94a}$$

(see (4.14) and (4.77)) and

$$|x_i| > \gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} \tag{4.94b}$$

(see (4.81)). If this is the case, then (4.93) must also hold true for

$$\gamma_k = \frac{l-1}{l^2 \epsilon_j^{\frac{l-1}{l}}} \tag{4.95a}$$

and

$$|x_i| = \gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} \tag{4.95b}$$

Combining (4.93), (4.95a), and (4.95b) and simplifying, we obtain

$$\left[\frac{(l-1)^2}{l^4 \epsilon_j^{\frac{l-1}{l}}}\right]^l > \left[\frac{(l-1)^2}{l^3(l+1)\epsilon_j} + \frac{\epsilon_j}{l+1}\right]^{l+1} \tag{4.96}$$

which is a contradiction because the right-hand side of (4.96) is greater than its left-hand side for all ϵ_j with $j \in \{0\} \cup \mathcal{I}_q$ and for all $l \in \{2,\ 3,\ \ldots\}$. Thus, for values of γ_k and $|x_i|$ given by 4.94a and (4.94b), respectively, we must have

$$\left(\frac{\gamma_k \frac{l}{l-1}}{l}\right)^l < \left(\frac{-|x_i| - \epsilon_j}{l+1}\right)^{l+1} \tag{4.97}$$

which implies that the trinomial equation in (4.82) has $(l-2)$ imaginary roots and 3 distinct real roots when l is even and $(l-1)$ imaginary roots and 2 distinct real roots when l is odd (see Problems 14 and 15 on p. 113 of [110]).

When l is even, the three distinct real roots are given by (4.89), and by

$$v_1 = \left\{ \gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} + \frac{\left[\gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} + \left(\frac{\gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}}}{|x_i|+\epsilon_j} \right)^{l+1} \right]^{l+1}}{|x_i|+\epsilon_j} \right\}^{l+1} \ldots \tag{4.98}$$

and

$$v_2 = \sqrt[l+1]{-\gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} - \frac{1}{\sqrt[l]{l+1}} (|x_i|+\epsilon_j)^{\frac{l+1}{l}} + \sqrt[l+1]{-\gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} - \frac{1}{\sqrt[l]{l+1}} (|x_i|+\epsilon_j)^{\frac{l+1}{l}} \sqrt[l+1]{-\gamma_k \frac{l-1}{l\sqrt[l]{\epsilon_j}} \ldots}}} \tag{4.99}$$

(see Sec. 10 of [99]). It can be shown that for values of γ_k and $|x_i|$ given by 4.94a and (4.94b), respectively, root v in (4.89) is always greater than $\sqrt[l]{\epsilon_j}$, while roots v_1 and v_2 in (4.98) and (4.99), respectively, are always less than $\sqrt[l]{\epsilon_j}$. When l is even, the two distinct real roots are given by v and v_1. For values of γ_k and $|x_i|$ given by 4.94a and (4.94b), respectively, root v in (4.89) is always greater than $\sqrt[l]{\epsilon_j}$, while root v_1 in (4.98) is always less than $\sqrt[l]{\epsilon_j}$.

When $l=2$, the equation in (4.82) has three roots given by (4.92), and by

$$v_1 = \left(\frac{-1+\jmath\sqrt{3}}{2}\right) c - \left(\frac{-1+\jmath\sqrt{3}}{2}\right)^2 \frac{a}{3c} \tag{4.100}$$

and

$$v_2 = \left(\frac{-1+\jmath\sqrt{3}}{2}\right)^2 c - \left(\frac{-1+\jmath\sqrt{3}}{2}\right) \frac{a}{3c} \tag{4.101}$$

(see Chapter 5 of [110]). For values of γ_k and $|x_i|$ given by 4.94a and (4.94b), respectively, we conclude that the roots in (4.92), (4.100), and (4.101) are all real (see (4.97)). Root v in (4.92) is always greater than $\sqrt{\epsilon_j}$, while roots v_1 and v_2 in (4.100) and (4.101), respectively, are always less than $\sqrt{\epsilon_j}$. When $l=3$, the equation in (4.82) has four roots given by (4.92), and by

$$v_1 = \frac{1}{2}\left[\sqrt{c-\frac{a}{3c}} - \sqrt{4\sqrt{\frac{1}{4}\left(c-\frac{a}{3c}\right)^2 - \gamma_k\frac{l-1}{l}} - \left(c-\frac{a}{3c}\right)}\right] \tag{4.102}$$

$$v_2 = \frac{1}{2}\left[-\sqrt{c-\frac{a}{3c}} + \sqrt{4\sqrt{\frac{1}{4}\left(c-\frac{a}{3c}\right)^2 - \gamma_k\frac{l-1}{l}} - \left(c-\frac{a}{3c}\right)}\right] \tag{4.103}$$

and

$$v_3 = \frac{1}{2}\left[-\sqrt{c-\frac{a}{3c}} - \sqrt{4\sqrt{\frac{1}{4}\left(c-\frac{a}{3c}\right)^2 - \gamma_k\frac{l-1}{l}} - \left(c-\frac{a}{3c}\right)}\right] \tag{4.104}$$

(see Chapter 5 of [110]). For values of γ_k and $|x_i|$ given by 4.94a and (4.94b), respectively, we conclude that the roots in (4.92) and (4.102) are real and those in (4.103) and (4.104) are imaginary (see (4.97)). Root v in (4.92) is always greater than $\sqrt[3]{\epsilon_j}$, while root v_1 in (4.102) is always less than $\sqrt[3]{\epsilon_j}$.

Lastly, from the above arguments, we conclude that roots v in (4.89) and (4.92)

are unique real roots greater than $\sqrt[l]{\epsilon_j}$ of the trinomial equation in (4.82). Therefore, they must also be the largest real roots of (4.82), which completes the proof. □

On the basis of Propositions 4.6 to 4.8, the PP of function $P_{\epsilon_j}(\boldsymbol{x})$ can be computed by using Algorithm 4.2.

Input: ϵ_j, $\boldsymbol{x} \in \mathbb{R}^n$, $\gamma_k < \frac{1}{\rho_{\epsilon_j}}$, $l \in \{2,\ 3,\ \ldots\}$, $\varepsilon_c > 0$
Output: $\boldsymbol{z} = \text{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x})\right]$
for $i = 1$ **to** n **do**
  if $l \in \{2,\ 3\}$ **then**
    Compute the root v given by (4.92);
  else
    $v = \sqrt[l+1]{-\gamma_k \frac{l-1}{l} - (-|x_i| - \epsilon_j)}$;
    repeat
      $v^{(0)} = v$;
      $v = \sqrt[l+1]{-\gamma_k \frac{l-1}{l} - (-|x_i| - \epsilon_j)v^{(0)}}$;
    until $||v - v^{(0)}|| \leq \varepsilon_c$;
  end
  Compute the PP z_i given by (4.81);
end
Compute the PP $\boldsymbol{z}$ given by (4.71);

Algorithm 4.2: Computation of the PP of $P_\epsilon(\boldsymbol{x})$

4.3.3 Projection onto the Feasible Set

From (1.11) and (4.10), the projection of a point $\bar{\boldsymbol{y}} \in \mathbb{R}^{n+m}$ onto set $\mathcal{K}_\delta$ is given by

$$\text{proj}_{\mathcal{K}_\delta}(\bar{\boldsymbol{y}}) = \arg \underset{\boldsymbol{y} \in \bigcap_{i=1}^{m+1} \mathcal{K}_{\delta,i}}{\text{minimize}} \ \|\boldsymbol{y} - \bar{\boldsymbol{y}}\| \tag{4.105}$$

where convex sets $\mathcal{K}_{\delta,i}$ for $i \in \mathcal{I}_m$ are hyperplanes of the form given in (4.8), convex set $\mathcal{K}_{\delta,i}$ for $i \in \{m+1\}$ is the closed ball under affine mapping given by (4.9), and

the variable $\boldsymbol{y} \in \mathbb{R}^{n+m}$ is defined in terms of variables $\boldsymbol{z}$ and $\boldsymbol{c}$ in (4.6) as

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{c} \end{bmatrix} \tag{4.106}$$

The convex feasibility problem associated with set $\mathcal{K}_\delta$ is closely related to the optimization problem in (4.105). It reduces to that of finding a common point of closed and convex sets, i.e.,

$$\text{find } \boldsymbol{y} \in \bigcap_{i=1}^{m+1} \mathcal{K}_{\delta,i} \tag{4.107}$$

Here we are searching for an efficient method for the solution of the aforementioned convex feasibility problem that can use matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ in matrix-vector operations only. This same method is applicable for computing the projection of a point onto set $\mathcal{K}_\delta$ because the problems in (4.105) and (4.107) can be shown to have the same solution in such a setting.

Let I_{m+1} denote a set of $m+1$ integers given by $\{1,\ \ldots,\ m,\ m+1\}$. Next, we partition set $\mathcal{I}_{m+1}$ into M blocks of indices as

$$\mathcal{I}_{m+1} = \mathcal{I}_{m+1}^1 \cup \mathcal{I}_{m+1}^2 \cup \cdots \cup \mathcal{I}_{m+1}^M \tag{4.108}$$

and let $\{t(k)\}_{k\in\mathbb{N}}$ denote a control sequence over the set $\{1,\ \ldots,\ M\}$, as is done in [2]. The problem in (4.107) can now be solved by applying an update formula of the form

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} + \lambda_k L_k \left[\sum_{i\in\mathcal{I}_{m+1}^{t(k)}} w_{i,k}\, \text{proj}_{\mathcal{K}_\delta^i}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)} \right] \tag{4.109}$$

where the sequence of weights $\{w_{i,k}\}_{i\in\mathcal{I}_{m+1}^{t(k)}}$ satisfies the conditions

1. $\{w_{i,k}\}_{i\in\mathcal{I}_{m+1}^{t(k)}} \subset [0,1]$
2. $\displaystyle\sum_{i\in\mathcal{I}_{m+1}^{t(k)}} w_{i,k} = 1$

The relaxation sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ satisfies the condition

$$\epsilon/L_k \leq \lambda_k \leq 2-\epsilon, \qquad \text{for } \epsilon \in (0,1)$$

and $L_k \geq 1$ is an extrapolation parameter (see [6, 29, 32, 33]). For example, if we let

$L_k = 1$ for all $k \in \mathbb{N}$ and let the control sequence $\{t(k)\}_{k\in\mathbb{N}}$ be given by

$$t(k) = k \pmod{m+1} + 1 \tag{4.110}$$

where the indices $\mathcal{I}_{m+1}^{t(k)}$ are defined by $\mathcal{I}_{m+1}^{t(k)} = t(k)$, then the update formula in (4.109) would correspond to that used in the AP method of [52].

Each solution point $\boldsymbol{y}^{(k)}$ in (4.109) can be computed efficiently because the projector onto each set $\mathcal{K}_{\delta,i}$ has a simple analytical solution. If we let $\boldsymbol{y}^{(k)}$ be of the form

$$\boldsymbol{y}^{(k)} = \begin{bmatrix} \boldsymbol{z}^{(k)} \\ \boldsymbol{c}^{(k)} \end{bmatrix} \tag{4.111}$$

then the projections for $i \in \mathcal{I}_m$ and $i \in \{m+1\}$ are given, respectively, by

$$\text{proj}_{\mathcal{K}_\delta^i}(\boldsymbol{y}^{(k)}) = \begin{bmatrix} \boldsymbol{z}^{(k)} \\ \boldsymbol{c}^{(k)} \end{bmatrix} + \frac{b_i + c_i^{(k)} - \boldsymbol{a}_i^T \boldsymbol{z}^{(k)}}{||\boldsymbol{a}_i||^2} \begin{bmatrix} \boldsymbol{a}_i \\ \boldsymbol{0} \end{bmatrix} \tag{4.112}$$

and

$$\text{proj}_{\mathcal{K}_\delta^i}(\boldsymbol{y}^{(k)}) = \begin{cases} \begin{bmatrix} \boldsymbol{z}^{(k)} \\ \boldsymbol{c}^{(k)} \end{bmatrix}, & ||\boldsymbol{c}^{(k)}|| \leq \delta \\ \begin{bmatrix} \boldsymbol{z}^{(k)} \\ \frac{1}{2}\left(1 + \frac{\delta}{||\boldsymbol{c}^{(k)}||}\right)\boldsymbol{c}^{(k)} \end{bmatrix}, & ||\boldsymbol{c}^{(k)}|| > \delta \end{cases} \tag{4.113}$$

(see pp. 398 and 447 of [16]). In addition, accelerated convergence of the sequence $\left\{\boldsymbol{y}^{(k)}\right\}_{k\in\mathbb{N}}$ in (4.109) is achieved by using an extrapolation parameter L_k of the form

$$L_k = \begin{cases} \dfrac{\sum_{i\in\mathcal{I}_{m+1}^{t(k)}} w_{i,k} ||\,\text{proj}_{\mathcal{K}_\delta^i}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)}||^2}{||\sum_{i\in\mathcal{I}_{m+1}^{t(k)}} w_{i,k}\,\text{proj}_{\mathcal{K}_\delta^i}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)}||^2}, & \boldsymbol{y}^{(k)} \notin \bigcap\limits_{i\in\mathcal{I}_{m+1}^{t(k)}} \mathcal{K}_\delta^i \\ 1, & \text{otherwise} \end{cases} \tag{4.114}$$

(see Secs. IV and V of [32]). Thus, by using (4.112) to (4.114), we conclude that the update formula in (4.109) can be computed efficiently.

For the case of orthogonal ensembles, matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ can be used in matrix-vector operations only during computation of the update formula in (4.109). This is achieved when the index set $\mathcal{I}_{m+1}$ in (4.108) is appropriately partitioned. We use

$M = 2$ with $\mathcal{I}^1_{m+1} = \{1,\ 2,\ \ldots,\ m\}$ and $\mathcal{I}^2_{m+1} = \{m+1\}$ in (4.108). The weights $w_{i,k}$ are of the form

$$w_{i,k} = \begin{cases} 1/m, & \text{for } i \in \mathcal{I}^1_{m+1} \\ 1, & \text{for } i \in \mathcal{I}^2_{m+1} \end{cases} \tag{4.115}$$

and the control sequence $\{t(k)\}_{k\in\mathbb{N}}$ is defined by

$$t(k) = k \pmod{M} + 1 \tag{4.116}$$

Consider the update formula when $\mathcal{I}^{t(k)}_{m+1}$ reduces to the block of indices $\mathcal{I}^1_{m+1}$. From (4.112) and (4.115), we obtain

$$\begin{aligned} \sum_{i\in\mathcal{I}^1_{m+1}} w_{i,k} ||\operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)}||^2 &= \frac{1}{m} \sum_{i\in\mathcal{I}^1_{m+1}} \left\| \frac{b_i + c_i^{(k)} - \boldsymbol{a}_i^T \boldsymbol{z}^{(k)}}{||\boldsymbol{a}_i||^2} \begin{bmatrix} \boldsymbol{a}_i \\ \boldsymbol{0} \end{bmatrix} \right\|^2 \\ &= \frac{1}{m} \left\| \begin{bmatrix} \operatorname{diag}(\boldsymbol{d})^{-1}(\boldsymbol{b} + \boldsymbol{c}^{(k)} - \boldsymbol{A}\boldsymbol{z}^{(k)}) \\ \boldsymbol{0} \end{bmatrix} \right\|^2 \end{aligned} \tag{4.117}$$

and

$$\begin{aligned} \left\| \sum_{i\in\mathcal{I}^1_{m+1}} w_{i,k} \operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)} \right\|^2 &= \left\| \frac{1}{m} \sum_{i\in\mathcal{I}^1_{m+1}} \frac{b_i + c_i^{(k)} - \boldsymbol{a}_i^T \boldsymbol{z}^{(k)}}{||\boldsymbol{a}_i||^2} \begin{bmatrix} \boldsymbol{a}_i \\ \boldsymbol{0} \end{bmatrix} \right\|^2 \\ &= \left\| \frac{1}{m} \begin{bmatrix} \boldsymbol{A}^T \left\{ \operatorname{diag}(\boldsymbol{d})^{-2}(\boldsymbol{b} + \boldsymbol{c}^{(k)} - \boldsymbol{A}\boldsymbol{z}^{(k)}) \right\} \\ \boldsymbol{0} \end{bmatrix} \right\|^2 \end{aligned} \tag{4.118}$$

where $\boldsymbol{d}$ denotes a column vector of length m which is formed by collecting the rows of matrix $\boldsymbol{A}$ as

$$\boldsymbol{d} = \begin{bmatrix} ||\boldsymbol{a}_1|| & ||\boldsymbol{a}_2|| & \cdots & ||\boldsymbol{a}_m|| \end{bmatrix}^T \tag{4.119}$$

Because matrix $\boldsymbol{A}$ is orthonormal when orthogonal ensembles are used, vector $\boldsymbol{d}$ reduces to a column vector of m ones. Therefore, by using (4.117) and (4.118), the parameter L_k in (4.114) can be written as

$$L_k = \begin{cases} \dfrac{\frac{1}{m} \left\| \boldsymbol{b} + \boldsymbol{c}^{(k)} - \boldsymbol{A}\boldsymbol{z}^{(k)} \right\|^2}{\left\| \frac{1}{m} \boldsymbol{A}^T \left(\boldsymbol{b} + \boldsymbol{c}^{(k)} - \boldsymbol{A}\boldsymbol{z}^{(k)} \right) \right\|^2}, & \boldsymbol{y}^{(k)} \notin \bigcap\limits_{i\in\mathcal{I}^1_{m+1}} \mathcal{K}^i_\delta \\ 1, & \text{otherwise} \end{cases} \tag{4.120}$$

and from (4.112), (4.115), and (4.120), the update formula in (4.109) reduces to

$$\begin{aligned}\boldsymbol{y}^{(k+1)} &= \boldsymbol{y}^{(k)} + \lambda_k L_k \left[\sum_{i \in \mathcal{I}^1_{m+1}} w_{i,k} \operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)} \right] \\ &= \begin{bmatrix} \boldsymbol{z}^{(k)} \\ \boldsymbol{c}^{(k)} \end{bmatrix} + \lambda_k L_k \frac{1}{m} \begin{bmatrix} \boldsymbol{A}^T \left(\boldsymbol{b} + \boldsymbol{c}^{(k)} - \boldsymbol{A}\boldsymbol{z}^{(k)} \right) \\ \boldsymbol{0} \end{bmatrix}\end{aligned} \tag{4.121}$$

when the block of indices $\mathcal{I}^1_{m+1}$ is used.

Consider now the update formula when $\mathcal{I}^{t(k)}_{m+1}$ reduces to the block of indices $\mathcal{I}^2_{m+1}$. From (4.115), we have

$$\sum_{i \in \mathcal{I}^2_{m+1}} w_{i,k} || \operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)}||^2 = || \sum_{i \in \mathcal{I}^2_{m+1}} w_{i,k} \operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)}||^2$$

which implies that the parameter L_k in (4.114) can be written as

$$\begin{aligned} L_k &= \begin{cases} \dfrac{\sum_{i \in \mathcal{I}^2_{m+1}} w_{i,k} || \operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)}||^2}{|| \sum_{i \in \mathcal{I}^2_{m+1}} w_{i,k} \operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)}||^2}, & \boldsymbol{y}^{(k)} \notin \bigcap\limits_{i \in \mathcal{I}^2_{m+1}} \mathcal{K}^i_\delta \\ 1, & \text{otherwise} \end{cases} \\ &= 1 \end{aligned} \tag{4.122}$$

Therefore, from (4.113) and (4.122), the update formula in (4.109) reduces to

$$\begin{aligned}\boldsymbol{y}^{(k+1)} &= \boldsymbol{y}^{(k)} + \lambda_k L_k \left[\sum_{i \in \mathcal{I}^2_{m+1}} w_{i,k} \operatorname{proj}_{\mathcal{K}^i_\delta}(\boldsymbol{y}^{(k)}) - \boldsymbol{y}^{(k)} \right] \\ &= \begin{bmatrix} \boldsymbol{z}^{(k)} \\ \boldsymbol{c}^{(k)} \end{bmatrix} + \lambda_k \begin{cases} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, & ||\boldsymbol{c}^{(k)}|| \le \delta \\ \begin{bmatrix} \boldsymbol{0} \\ \left\{ \frac{1}{2} \left(1 + \frac{\delta}{||\boldsymbol{c}^{(k)}||} \right) - 1 \right\} \boldsymbol{c}^{(k)} \end{bmatrix}, & ||\boldsymbol{c}^{(k)}|| > \delta \end{cases}\end{aligned} \tag{4.123}$$

when the block of indices $\mathcal{I}^2_{m+1}$ is used.

In summary, we conclude from (4.121) and (4.123) that the computation can be carried out by using fast algorithms for matrix-vector products in the case of

orthogonal ensembles. Thus, the update formula in (4.109) is suitable for solving large-scale recovery problems.

Lastly, consider the solution of the optimization problem in (4.105). As stated in the following theorem, the update formula in (4.109) can be used for computing not only a feasible point but also the projection of a point onto the feasible set.

Theorem 4.1 (Feasible Point and Projection of a Point)**.** *Let* $\{\boldsymbol{z}^{(k)}\}_{k\in\mathbb{N}}$ *denote a sequence obtained from sequence* $\{\boldsymbol{y}^{(k)}\}_{k\in\mathbb{N}}$ *in* (4.109) *by using the variable transformation in* (4.111)*. The sequence* $\{\boldsymbol{z}^{(k)}\}_{k\in\mathbb{N}}$ *converges to* $\text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z})$ *for every* $\boldsymbol{z}\in\mathbb{R}^n$*.*

Proof. Let $\{\bar{\boldsymbol{y}}^{(k)}\}_{k\in\mathbb{N}}$ denote a sequence obtained from sequence $\{\boldsymbol{y}^{(k)}\}_{k\in\mathbb{N}}$ in (4.109) when $L_k = 1$ for all $k\in\mathbb{N}$ and when the control sequence is given by (4.110). In addition, let $\{\bar{\boldsymbol{z}}^{(k)}\}_{k\in\mathbb{N}}$ denote a sequence obtained from $\{\bar{\boldsymbol{y}}^{(k)}\}_{k\in\mathbb{N}}$ by using the variable transformation in (4.111).

From Proposition 4.1 and (4.112) and (4.113), we conclude that the update formula associated with $\{\bar{\boldsymbol{z}}^{(k)}\}_{k\in\mathbb{N}}$ corresponds to that used in an AP method for finding a point in the intersection of linear varieties. Thus, sequence $\{\bar{\boldsymbol{z}}^{(k)}\}_{k\in\mathbb{N}}$ converges to $\text{proj}_{\bigcap_{i=1}^m \mathcal{K}_{\delta,i}}(\boldsymbol{z})$ for every $\boldsymbol{z}\in\mathbb{R}^n$ (see Corollary 2, pp. 50 of [114]). Now, if sequence $\{\bar{\boldsymbol{z}}^{(k)}\}_{k\in\mathbb{N}}$ converges to a point $\bar{\boldsymbol{z}}\in\bigcap_{i=1}^m \mathcal{K}_{\delta,i}$, then sequence $\{\boldsymbol{z}^{(k)}\}_{k\in\mathbb{N}}$ must also converge to that same point $\bar{\boldsymbol{z}}$. This can be shown by combining Lemma 4.1 (iv) and Proposition 2.2 (iii) of [33] into the proof of Theorem 4.2 of the same reference (see Theorem 4.1 of [33] for a similar result). Based on the above arguments, we conclude that $\{\boldsymbol{z}^{(k)}\}_{k\in\mathbb{N}}$ converges to $\text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z})$ for every $\boldsymbol{z}\in\mathbb{R}^n$, which completes the proof. □

On the basis of Theorem 4.1, the projection of a point $\boldsymbol{z}\in\mathbb{R}^n$ onto set $\mathcal{K}_\delta$ can be computed by using Algorithm 4.3.

4.3.4 Convergence Analysis

Consider the case where the update formula in (1.14) is applied for the minimization of function $F(\boldsymbol{x})$ and the problem in (1.13) is solved approximately. A requirement for the convergence of such inexact variant of the PP method to a minimizer of $F(\boldsymbol{x})$ is that the errors entailed by the approximate solution are summable (see p. 880 of [94] and p. 732 of [30]). On the basis of these convergence results, a necessary condition for the convergence of sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.63) to a point in the solution set $\mathcal{S}_{\epsilon_j,\delta}$ for all $j\in\{0\}\cup\mathcal{I}_q$ is that the magnitude of the error vector $\boldsymbol{e}_{\epsilon_j}^{(k)}$ in (4.61) is bounded and

Input: $\delta \geq 0, \varepsilon_c > 0, \boldsymbol{z} \in \mathbb{R}^n, \boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m, \lambda_k > 0$
Output: $\text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z})$
$\boldsymbol{c}^{(0)} = \boldsymbol{A}\boldsymbol{z} - \boldsymbol{b}$;
$\boldsymbol{z}^{(0)} = \boldsymbol{z}$;
$k = -1$;
repeat
  $k = k + 1$;
  Compute the control sequence $t(k)$ given by (4.116);
  if $t(k) = 1$ **then**
    Compute the solution point $\boldsymbol{y}^{(k+1)}$ given by (4.121);
  else if $t(k) = 2$ **then**
    Compute the solution point $\boldsymbol{y}^{(k+1)}$ given by (4.123);
  end
until $||\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}|| \leq \varepsilon_c$;
$\text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z}) = \boldsymbol{z}^{(k+1)}$;

Algorithm 4.3: Projection onto $\mathcal{K}_\delta$

the resulting error sequence $\{||\boldsymbol{e}_{\epsilon_j}^{(k)}||\}_{k\in\mathbb{N}}$ is monotonically decreasing and summable. As stated in the following lemma, the errors entailed by the proposed update formula have the aforementioned properties.

Lemma 4.4 (Boundedness and Summability of the Error)**.** *If $\gamma_k < 1/\rho_{\epsilon_j}$ for all $j \in \{0\} \cup \mathcal{I}_q$, the following properties apply to the error vector $\boldsymbol{e}_{\epsilon_j}^{(k)}$ in (4.61):*

1. *The magnitude of the error is bounded as*

$$0 \leq ||\boldsymbol{e}_{\epsilon_j}^{(k)}|| \leq \sqrt{2\gamma_k n \frac{p}{\epsilon_j^{1-p}}}, \qquad \text{for all } k \in \mathbb{N},\ j \in \{0\} \cup \mathcal{I}_q \tag{4.124}$$

2. *The error sequence $\{||\boldsymbol{e}_{\epsilon_j}^{(k)}||\}_{k\in\mathbb{N}}$ defined by the update formula in (4.59) is a monotonically decreasing and summable sequence.*

Proof. We have

$$\begin{aligned} ||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k)}||^2 &= ||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)} + \boldsymbol{x}_{\epsilon_j}^{(k+1)} - \boldsymbol{z}_{\epsilon_j}^{(k)}||^2 \\ &= ||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)}||^2 + ||\boldsymbol{x}_{\epsilon_j}^{(k+1)} - \boldsymbol{z}_{\epsilon_j}^{(k)}||^2 - 2(\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)})^T(\boldsymbol{z}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)}) \end{aligned} \tag{4.125}$$

for all $k \in \mathbb{N}$. The projection of $\boldsymbol{z}_{\epsilon_j}^{(k)}$ onto set $\mathcal{K}_\delta$ has the property

$$\left[\boldsymbol{z}_{\epsilon_j}^{(k)} - \text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z}_{\epsilon_j}^{(k)})\right]^T \left[\boldsymbol{x} - \text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z}_{\epsilon_j}^{(k)})\right] \leq 0, \qquad \text{for all } \boldsymbol{x} \in \mathcal{K}_\delta \tag{4.126}$$

(see Theorem E.9.1.0.2 of [36]). By using (4.63) and (4.126), we have

$$(\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)})^T(\boldsymbol{z}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k+1)}) \leq 0, \qquad \text{for all } k \in \mathbb{N} \tag{4.127}$$

Combining (4.125) and (4.127), we obtain the inequality

$$||\boldsymbol{x}_{\epsilon_j}^{(k+1)} - \boldsymbol{z}_{\epsilon_j}^{(k)}|| \leq ||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k)}||, \qquad \text{for all } k \in \mathbb{N} \tag{4.128}$$

From Corollary 4.1, function $P_{\epsilon_j}(\boldsymbol{x})$ is Lipschitz continuous with constant κ_{ϵ_j} for all $j \in \{0\} \cup \mathcal{I}_q$ and, therefore, the Euclidean distance between a PP $\boldsymbol{z}_{\epsilon_j}^{(k)}$ and the prox-center $\boldsymbol{x}_{\epsilon_j}^{(k)}$ is bounded from above as

$$||\boldsymbol{z}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k)}|| \leq 2\gamma_k\kappa_{\epsilon_j}, \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.129}$$

(see Lemma 2 of [56]). Thus, combining (4.21), (4.61), (4.128), and (4.129), we obtain (4.124).

Now from (4.63), (4.64), and (4.69a), we obtain

$$\begin{aligned} ||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k)}|| &= \left\|\text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z}_{\epsilon_j}^{(k-1)}) - \text{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})\right]\right\| \\ ||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k-1)}|| &= \left\|\text{proj}_{\mathcal{K}_\delta}(\boldsymbol{z}_{\epsilon_j}^{(k-1)}) - \text{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k-1)})\right]\right\| \end{aligned}, \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.130}$$

Furthermore, the inequality in (4.128) can be written as

$$||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k-1)}|| \leq ||\boldsymbol{x}_{\epsilon_j}^{(k-1)} - \boldsymbol{z}_{\epsilon_j}^{(k-1)}||, \qquad \text{for all } k \in \mathbb{N} \tag{4.131}$$

Therefore, from (1.11), (1.15), (4.130), and (4.131), we must have

$$||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k)}|| \leq ||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k-1)}||, \qquad \text{for all } k \in \mathbb{N} \tag{4.132}$$

and by combining (4.128) and (4.132), we obtain the inequality

$$||\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{z}_{\epsilon_j}^{(k-1)}|| \geq ||\boldsymbol{x}_{\epsilon_j}^{(k+1)} - \boldsymbol{z}_{\epsilon_j}^{(k)}||, \qquad \text{for all } k \in \mathbb{N}$$

or by using (4.61), the inequality

$$||\boldsymbol{e}_{\epsilon_j}^{(k)}|| \geq ||\boldsymbol{e}_{\epsilon_j}^{(k+1)}||, \qquad \text{for all } k \in \mathbb{N} \tag{4.133}$$

Hence, from (4.124) and (4.133), we conclude that the error sequence $\{||\boldsymbol{e}_{\epsilon_j}^{(k)}||\}_{k\in\mathbb{N}}$ is a bounded monotonically decreasing sequence. Thus, $\{||\boldsymbol{e}_{\epsilon_j}^{(k)}||\}_{k\in\mathbb{N}}$ is convergent (see monotonic sequence theorem, p. 710 of [100]). By following a similar approach, it can be shown that the related sequence $\{||\boldsymbol{e}_{\epsilon_j}^{(k+1)}||/||\boldsymbol{e}_{\epsilon_j}^{(k)}||\}_{k\in\mathbb{N}}$ converges to some limit $\ell < 1$ and, therefore, sequence $\{||\boldsymbol{e}_{\epsilon_j}^{(k)}||\}_{k\in\mathbb{N}}$ is summable (see ratio test, p. 743 of [100]), which completes the proof. □

Now let $\mathrm{T}_{\epsilon,\delta}$ denote the operator associated with the subdifferential mapping of function $\widetilde{F}_{\epsilon,\delta}(\boldsymbol{x})$ and let $\mathcal{S}_{\epsilon,\delta}$ denote the set of minimizers of $\widetilde{F}_{\epsilon,\delta}(\boldsymbol{x})$. If we suppose that $\mathrm{T}_{\epsilon,\delta}$ is monotone and the sequence of prox-parameters $\{\gamma_k\}_{k\in\mathbb{N}}$ defined by (4.52) is forced to assume a value above zero, i.e., $\inf\{\gamma_k\}_{k\in\mathbb{N}} > 0$, then the summability of the errors entailed by the approximate solution of (4.48) is sufficient for the convergence of inexact variants of the PP method to a solution point $\boldsymbol{x}_{\epsilon,\delta}^{*} \in \mathcal{S}_{\epsilon,\delta}$ (see Theorem 1 of [94]). In the case where $\mathrm{T}_{\epsilon,\delta}$ is η_ϵ-cohypomonotone, additional conditions for convergence are necessary such as (1) each solution point $\boldsymbol{x}_{\epsilon}^{(k)}$ in (4.52) is within a neighborhood of $\mathcal{S}_{\epsilon,\delta}$, (2) the initial point $\boldsymbol{x}_{\epsilon}^{(0)}$ is close enough to $\mathcal{S}_{\epsilon,\delta}$, (3) the errors entailed by the approximate solution are not only summable but smaller than the difference between the size of the neighborhood of $\mathcal{S}_{\epsilon,\delta}$ and the distance of $\boldsymbol{x}_{\epsilon}^{(0)}$ to $\mathcal{S}_{\epsilon,\delta}$, and (4) the sequence $\{\gamma_k\}_{k\in\mathbb{N}}$ is bounded away from η_ϵ (see Theorem 3.1 of [30]). On the basis of these results, the conditions for the convergence of the IPPP method to a point in $\mathcal{S}_{\epsilon_j,\delta}$ for all $j \in \{0\} \cup \mathcal{I}_q$ are stated in Theorem 4.2 below.

Theorem 4.2 (Convergence to Solution Set with Regularization Sequence)**.** *Let $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ be a sequence as described in Definition 4.3. Assume that the approximation in (4.42) holds true and let η_{ϵ_j} be given by (4.51). Consider applying the update formula in (4.59) for the solution of each problem in the sequence $\{(LP_{\epsilon_j,\delta})\}_{j\in\{0\}\cup\mathcal{I}_q}$ and let $\boldsymbol{x}_{\epsilon_0}^{(0)} \in \mathcal{K}_\delta$ denote a given initial point for the solution of problem $(LP_{\epsilon_0,\delta})$. For the case where $j = 0$, there exists a closed ball $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ in (4.24) such that the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.59) converges to a solution point $\boldsymbol{x}_{\epsilon_j,\delta}^{*} \in \mathcal{S}_{\epsilon_j,\delta}$ if the following conditions are satisfied:*

1. *The solution point $\boldsymbol{x}_{\epsilon_j}^{(k)}$ in (4.59) is such that*

$$\boldsymbol{x}_{\epsilon_j}^{(k)} \in B(\alpha, \mathcal{S}_{\epsilon_j,\delta}), \qquad \textit{for all } k \in \mathbb{N}$$

2. *The sequence of prox-parameters* $\{\gamma_k\}_{k\in\mathbb{N}}$ *defined by* (4.59) *is such that*

$$2\eta_{\epsilon_j} < \inf\{\gamma_k\}_{k\in\mathbb{N}} \qquad \textit{and} \qquad \sup\{\gamma_k\}_{k\in\mathbb{N}} < \frac{1}{\rho_{\epsilon_j}}$$

3. *The distance* $d(\boldsymbol{x}_{\epsilon_j}^{(0)}, \mathcal{S}_{\epsilon_j,\delta})$ *in* (4.26) *is such that*

$$d(\boldsymbol{x}_{\epsilon_j}^{(0)}, \mathcal{S}_{\epsilon_j,\delta}) < \frac{4}{5}\alpha$$

4. *The regularization parameter* ϵ_j *is large enough so that*

$$\sqrt{2\gamma_k n \frac{p}{\epsilon_j^{1-p}}} \approx 0, \qquad \textit{for all } k \in \mathbb{N}$$

Furthermore, for the case where $j \in \mathcal{I}_q$, *there exists a closed ball* $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ *in* (4.24) *such that the sequence* $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ *in* (4.59) *converges to a solution point* $\boldsymbol{x}_{\epsilon_j,\delta}^* \in \mathcal{S}_{\epsilon_j,\delta}$ *if Conditions 1 and 2 above are satisfied and, in addition, the upper bound* ν *in Property 3 of Definition 4.3 is small enough so that*

$$\sum_{k\in\mathbb{N}} ||\boldsymbol{e}_{\epsilon_j}^{(k)}|| < \frac{1}{2}\left(\frac{4}{5}\alpha - \nu^p \frac{\epsilon_j^{1-p}}{p}\right) \tag{4.134}$$

Proof. Using an approach similar to that in the proof of Theorem 4.3 of [30], the convergence of sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.59) to a solution point $\boldsymbol{x}_{\epsilon_j,\delta}^* \in \mathcal{S}_{\epsilon_j,\delta}$ for all $j \in \{0\}\cup\mathcal{I}_q$ can be established by verifying conditions (i), (iii), and (iv-vii) in Theorem 3.1 of [30] as follows.

There exists a closed ball $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ in (4.24) such that the operator $\mathrm{T}_{\epsilon_j,\delta}$ associated with function $F_{\epsilon_j,\delta}(\boldsymbol{x})$ in (4.4) is η_{ϵ_j}-cohypomonotone on $B(\alpha, \mathcal{S}_{\epsilon_j,\delta})$ for all $j \in \{0\} \cup \mathcal{I}_q$ (see Lemma 4.3). Thus, condition (i) in Theorem 3.1 of [30] holds true for η_{ϵ_j} given by (4.51) and for all $j \in \{0\} \cup \mathcal{I}_q$.

Let $\gamma_{\epsilon_j} = \inf\{\gamma_k\}_{k\in\mathbb{N}}$ for the corresponding update formula applied to the problem sequence $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$. Condition (iii) and (iv) in Theorem 3.1 of [30] read

$$\gamma_{\epsilon_j} > \eta_{\epsilon_j}, \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.135}$$

and

$$\varepsilon \leq \frac{1}{1-\eta_{\epsilon_j}/\gamma_{\epsilon_j}} \leq 2-\varepsilon, \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.136}$$

respectively, for some $\varepsilon \in (0,1)$. The inequality in (4.135) and the one in (4.136) for the case where $\varepsilon \to 0$ are implied by Condition 2 above.

Conditions (v) and (vi) in Theorem 3.1 of [30] read

$$d(\boldsymbol{x}_{\epsilon_j}^{(0)}, \mathcal{S}_{\epsilon_j,\delta}) < \frac{4-2\varepsilon}{5-2\varepsilon}\alpha, \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.137}$$

and

$$\sum_{k\in\mathbb{N}} ||\boldsymbol{e}_{\epsilon_j}^{(k)}|| < \frac{\frac{4-2\varepsilon}{5-2\varepsilon}\alpha - d(\boldsymbol{x}_{\epsilon_j}^{(0)}, \mathcal{S}_{\epsilon_j,\delta})}{2-\varepsilon}, \qquad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.138}$$

respectively, for some $\varepsilon \in (0,1)$. For the case where $j = 0$ and $\varepsilon \to 0$, the inequality in (4.137) is implied by Condition 3. Furthermore, by combining Condition 4 and (4.124), we obtain $\sum_{k\in\mathbb{N}} ||\boldsymbol{e}_{\epsilon_j}^{(k)}|| \approx 0$ (see Lemma 4.4). Thus, the inequality in (4.138) holds true. For the case where $j \in \mathcal{I}_q$ and $\varepsilon \to 0$, by combining (4.26), (4.29), and (4.66), we obtain $d(\boldsymbol{x}_{\epsilon_j}^{(0)}, \mathcal{S}_{\epsilon_j,\delta}) < \nu^p \epsilon_j^{1-p}/p$ (see Lemma 4.1). Thus, the inequalities in (4.137) and (4.138) are implied by (4.134).

Lastly, Condition (vii) in Theorem 3.1 of [30] is implied by Condition 1 above for all $j \in \{0\} \cup \mathcal{I}_q$, which completes the proof. □

4.3.5 Accelerated Convergence with Two-Step Method

The convergence of the IPPP method can be accelerated by employing a two-step method as detailed below. By applying the update-formula in (1.20) with $F(\boldsymbol{x}^{(k)}) = \psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}^{(k)})$ and $\alpha_k = \gamma_k$, we obtain a variant of the update formula in (4.52) given by

$$\boldsymbol{x}_{\epsilon_j}^{(k+1)} = \boldsymbol{x}_{\epsilon_j}^{(k)} - \gamma_k \boldsymbol{\nabla}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}^{(k)}) + \beta_k \left(\boldsymbol{x}_{\epsilon_j}^{(k)} - \boldsymbol{x}_{\epsilon_j}^{(k-1)}\right), \quad \text{for all } j \in \{0\} \cup \mathcal{I}_q \tag{4.139}$$

where $0 \leq \beta_k < 1$, $\boldsymbol{x}_{\epsilon_j}^{(-1)} = \boldsymbol{x}_{\epsilon_j}^{(0)}$ and the gradient vector $\boldsymbol{\nabla}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})$ is given by

$$\boldsymbol{\nabla}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)}) = \frac{1}{\gamma_k}\left[\boldsymbol{x}_{\epsilon_j}^{(k)} - \text{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{x}_{\epsilon_j}^{(k)} - \gamma_k\, \boldsymbol{g}_{\epsilon_j}^{(k+1)}\right)\right] \tag{4.140}$$

where $\boldsymbol{g}_{\epsilon_j}^{(k+1)}$ is of the form in (4.54) (see Lemma 4.2 and Proposition 4.4). The update formula in (4.139) corresponds to a two-step method for the solution of each problem

in the sequence $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$. The rate of convergence of sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.139) can be improved over that in (4.52) when parameter β_k is appropriately chosen. If we let β_k be given by (1.27) and if we suppose that (1) the ME $\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x})$ in (4.48) is convex and differentiable with a Lipschitz continuous gradient on the basis of Lemma 4.2 and (2) the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.52) converges to a solution point $\boldsymbol{x}_{\epsilon_j,\delta}^{*}\in\mathcal{S}_{\epsilon_j,\delta}$ for all $j\in\{0\}\cup\mathcal{I}_q$, on the basis of Theorem 4.2, then the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.139) must also converge to a solution point at a rate given by (1.21) (see Scheme 2.2.9 of [82] and Theorem 1 of [81]).

Now, by following a similar approach as that in Sec. 4.3, we have an inexact variant of the update formula in (4.139) given by

$$\boldsymbol{x}_{\epsilon_j}^{(k+1)}=\boldsymbol{x}_{\epsilon_j}^{(k)}-\gamma_k\boldsymbol{\nabla}\tilde{\psi}_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})+\beta_k\left(\boldsymbol{x}_{\epsilon_j}^{(k)}-\boldsymbol{x}_{\epsilon_j}^{(k-1)}\right),\quad\text{for all } j\in\{0\}\cup\mathcal{I}_q \tag{4.141}$$

where β_k is given by (1.27), the inexact gradient $\boldsymbol{\nabla}\tilde{\psi}_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})$ is given by

$$\boldsymbol{\nabla}\tilde{\psi}_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})=\frac{1}{\gamma_k}\left[\boldsymbol{x}_{\epsilon_j}^{(k)}-\mathrm{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{x}_{\epsilon_j}^{(k)}-\gamma_k\,\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)}\right)\right] \tag{4.142}$$

and $\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)}$ is of the form in (4.60). It has been demonstrated in [35] that the optimal convergence rate of two-step methods of the form in (4.141) is preserved when the inexact gradient $\tilde{\psi}_{\gamma_k,\epsilon_j}(\boldsymbol{x})$ is computed only up to a small uniformly bounded error. Thus, the rate of convergence of sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.141) can be improved over that in (4.59). On the basis of such results, we now show that the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.141) has an optimal convergence rate as stated in the following proposition.

Proposition 4.9 (Optimal Convergence Rate)**.** *Let $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ be a sequence as described in Definition 4.3 and let $\gamma_k\in(0,1/r_{\epsilon_j})$. For all $j\in\{0\}\cup\mathcal{I}_q$, assume that (1) the ME $\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x})$ in (4.48) is convex and differentiable with a Lipschitz continuous gradient and (2) the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.59) converges to a solution point $\boldsymbol{x}_{\epsilon_j,\delta}^{*}\in\mathcal{S}_{\epsilon_j,\delta}$. Under these circumstances, there exists a closed ball $B(\alpha,\mathcal{S}_{\epsilon_j,\delta})$ in (4.24) such that if (1) parameter β_k is given by (1.27) and (2) the solution point $\boldsymbol{x}_{\epsilon_j}^{(k)}$ is such that $\boldsymbol{x}_{\epsilon_j}^{(k)}\in B(\alpha,\mathcal{S}_{\epsilon_j,\delta})$ for all $k\in\mathbb{N}$, then the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.141) converges to a solution point $\boldsymbol{x}_{\epsilon_j,\delta}^{*}\in\mathcal{S}_{\epsilon_j,\delta}$ for all $j\in\{0\}\cup\mathcal{I}_q$ at a rate given by (1.21).*

Proof. From (4.140) and (4.142), we obtain

$$\begin{aligned}\left\|\widetilde{\boldsymbol{\nabla}}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})-\boldsymbol{\nabla}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})\right\| &= \\ \left\|\frac{1}{\gamma_k}\left[\mathrm{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{x}_{\epsilon_j}^{(k)}-\gamma_k\,\boldsymbol{g}_{\epsilon_j}^{(k+1)}\right)-\mathrm{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{x}_{\epsilon_j}^{(k)}-\gamma_k\,\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)}\right)\right]\right\| & \\ \leq \left\|\frac{1}{\gamma_k}\left(\boldsymbol{x}_{\epsilon_j}^{(k)}-\gamma_k\boldsymbol{g}_{\epsilon_j}^{(k+1)}-\boldsymbol{x}_{\epsilon_j}^{(k)}+\gamma_k\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)}\right)\right\| & \\ \leq \left\|\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)}-\boldsymbol{g}_{\epsilon_j}^{(k+1)}\right\| & \\ \leq \left\|\tilde{\boldsymbol{g}}_{\epsilon_j}^{(k+1)}\right\|+\left\|\boldsymbol{g}_{\epsilon_j}^{(k+1)}\right\| &\end{aligned} \tag{4.143}$$

for all $j\in\{0\}\cup\mathcal{I}_q$ and for all $k\in\mathbb{N}$, where the first inequality follows from the non-expansiveness of the projection operator (see Theorem E.9.3.0.1 of [36]) and the last one follows from the triangle inequality.

On the basis of Corollary 4.1 and Lemma 4.4, function $P_{\epsilon_j}(\boldsymbol{x})$ is Lipschitz continuous and $||\boldsymbol{e}_{\epsilon_j}^{(k)}||$ is bounded. Thus, for all $j\in\{0\}\cup\mathcal{I}_q$ and for all $k\in\mathbb{N}$, we obtain

$$\left\|\widetilde{\boldsymbol{\nabla}}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})-\boldsymbol{\nabla}\psi_{\gamma_k,\epsilon_j}(\boldsymbol{x}_{\epsilon_j}^{(k)})\right\| \leq 2n\frac{p}{\epsilon_j^{1-p}} \tag{4.144}$$

by combining (1.18), (4.21), and (4.143). Therefore, the gradient approximation error is bounded and convergence of the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ in (4.141) to a solution point $\boldsymbol{x}_{\epsilon_j,\delta}^{*}\in\mathcal{S}_{\epsilon_j,\delta}$ at a rate given by (1.21) follows from Theorem 2.2 of [35]. □

From Proposition 4.5, the update formula in (4.141) can be written as

$$\boldsymbol{x}_{\epsilon_j}^{(k+1)}=\mathrm{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{z}_{\epsilon_j}^{(k)}\right)+\beta_k\left(\boldsymbol{x}_{\epsilon_j}^{(k)}-\boldsymbol{x}_{\epsilon_j}^{(k-1)}\right),\qquad \text{for all } j\in\{0\}\cup\mathcal{I}_q \tag{4.145}$$

where $\boldsymbol{z}_{\epsilon_j}^{(k)}$ is given by (4.64). Therefore, the update formula in (4.145) corresponds to an accelerated version of the IPPP method in (4.63) because (1) such update formula boils down to the iterative computation of a PP of function $P_{\epsilon_j}(\boldsymbol{x})$ followed by the projection of such a point into the feasible set $\mathcal{K}_\delta$ and (2) the sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ involved in the computation has an optimal convergence rate on the basis of Proposition 4.9. Thus, the proposed update formula in (4.145) defines what may be referred to as the *fast* iterative proximal-point projection (FIPPP) method. This method with continuation is described in Algorithm 4.4.

Input: $\boldsymbol{x}^{(0)} \in \mathcal{K}_\delta$, $\varepsilon_c > 0$, $\{\epsilon_j\}_{j \in \mathcal{I}_q}$, $\gamma_k \in (0, 1/r_{\epsilon_j})$
Output: $\boldsymbol{x}^*_{\epsilon_q} \in \mathcal{S}_{\epsilon_q,\delta}$
$\boldsymbol{x}^*_{\epsilon_{-1}} = \boldsymbol{x}^{(0)}$;
for $j = 0$ **to** q **do**
  $t_0 = 1$;
  $\boldsymbol{x}^{(0)}_{\epsilon_j} = \boldsymbol{x}^*_{\epsilon_{j-1}}$;
  $\boldsymbol{x}^{(-1)}_{\epsilon_j} = \boldsymbol{x}^{(0)}_{\epsilon_j}$;
  $k = -1$;
  repeat
    $k = k + 1$;
    Compute the PP of $P_{\epsilon_j}(\boldsymbol{x}^{(k)}_{\epsilon_j})$ (see Algorithm 4.2);
    $\boldsymbol{z}^{(k)}_{\epsilon_j} = \text{prox}_{\gamma_k}\left[P_{\epsilon_j}(\boldsymbol{x}^{(k)}_{\epsilon_j})\right]$;
    Compute t_{k+1} given by (1.28);
    Compute β_k given by (1.27);
    Compute the projection of $\boldsymbol{z}^{(k)}_{\epsilon_j}$ onto set $\mathcal{K}_\delta$ (see Algorithm 4.3);
    $\boldsymbol{x}^{(k+1)}_{\epsilon_j} = \text{proj}_{\mathcal{K}_\delta}\left(\boldsymbol{z}^{(k)}_{\epsilon_j}\right) + \beta_k\left(\boldsymbol{x}^{(k)}_{\epsilon_j} - \boldsymbol{x}^{(k-1)}_{\epsilon_j}\right)$;
  until $||F_{\epsilon,\delta}(\boldsymbol{x}^{(k+1)}_{\epsilon_j}) - F_{\epsilon,\delta}(\boldsymbol{x}^{(k)}_{\epsilon_j})|| \leq \varepsilon_c$;
  $\boldsymbol{x}^*_{\epsilon_j} = \boldsymbol{x}^{(k+1)}_{\epsilon_j}$;
end

Algorithm 4.4: FIPPP Method with Continuation

4.4 Simulation Results

We now present simulation results to evaluate the capabilities of the proposed and corresponding competing methods in recovering realistic signals in a wide range of test problems with large dynamic range (DR) and scale. Signal-recovery methods were evaluated following the experimental protocol described in Sec. 1.3. The average ℓ_∞ reconstruction error, probability of perfect recovery (PPR), and minimum required fraction (MRF) were employed as reconstruction performance (RP) metrics. The difference between the Euclidean distance $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ and the estimate of the square root of the measurement noise energy δ was employed as a measurement consistency (MC) metric. The average CPU time in seconds and the average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ were employed as computational cost (CC)

metrics.

RP, MC, and CC metrics were computed by carrying out the recovery process 100 times using orthogonal measurement ensembles. The length of the signal of interest $\boldsymbol{x}^0$ was assumed to be $n = 1,048,576$. Each measurement vector $\boldsymbol{b}$ was generated by applying a renormalized discrete Fourier transform (DFT) matrix to $\boldsymbol{x}^0$. The length of vector $\boldsymbol{b}$ was assumed to be $m = n/8$ and the sparsity of $\boldsymbol{x}^0$ was assumed to be in the range $\frac{m}{100} \leq s \leq m$. The MRF, m/s, for perfect recovery was estimated by finding the minimum value of s in the aforementioned range where PPR = 1. The s nonzero values of $\boldsymbol{x}^0$ were generated as in (1.37) with parameter $\kappa \in \{1,\ 2,\ 4,\ 5\}$ which results in signals with DRs ranging from 20 dB to 100 dB. For instance, the absolute values of the nonzero entries of signals of 20 dB are distributed between 1 and 10 while signals of 100 dB have values distributed between 1 and 100,000. For noisy signals, each measurement vector $\boldsymbol{b}$ was obtained using a Gaussian vector $\boldsymbol{z}$ with $\sigma_{\boldsymbol{z}} = 1 \times 10^{-4}$ and perfect signal recovery was declared when $\nu = 5 \times 10^{-2}$ in (1.36). For noiseless signals, perfect recovery was declared when $\nu = 1 \times 10^{-3}$.

All experiments were run on a Dell Precision 670 workstation with two 3.2 GHz dual-core Intel Xeon processors and 4 Gb of RAM using the 64-bit Linux MATLAB Version 8.2.0.701 (R2013b). Software that is publicly available online was used for the competing methods.[1] We used the values suggested in [9, 11] for the several parameters of the software obtained. Among the competing methods mentioned in Fig. 1.5, the SPGL1 and NESTA methods were the only two methods capable of obtaining consistent results for the problems under consideration. Competing methods based on second-order solvers (SOSs) such as the iteratively reweighted least squares (IRWLS) [27], ℓ_1-Magic [18], and ℓ_1-LS [68] solvers suffered from numerical instabilities when solving the large linear system of equations involved in computing the Newton step. Competing methods based on FOSs, such as the gradient projection for sparse reconstruction (GPSR) [43], fast iterative shrinkage-thresholding algorithm (FISTA) [8], and difference-of-two-convex-functions (DC)-Family [46], do not entail the solution of large linear systems but they do require heuristics to find an appropriate value of λ for the problem (QP_λ) in (1.8). The recovery process carried out

[1] The codes for the competing methods were obtained from the respective author's Web pages:

Least absolute shrinkage and selection operator (LASSO) methods: Spectral projected-gradient ℓ_1-norm (SPGL1) from M. P. Friedlander at http://www.cs.ubc.ca/~mpf/spgl1/.

Basis pursuit (BP) methods: NESTA from E. J. Candès at http://www-stat.stanford.edu/~candes/nesta/.

with approximate values of λ was found to be unreliable for the hard test problems at hand.

4.4.1 Evaluation of proposed BP method

We carried out simulations to assess the convergence of the proposed method to a solution when appropriate initialization and supporting sequences are employed. RP and CC metrics were evaluated for different forms of the SPF and two-step acceleration. Algorithm 4.3 was used with $\varepsilon_c = 1 \times 10^{-6}$ and for all $k \in \mathbb{N}$ we used $\lambda_k = 1$ and $\lambda_k = 2 - 10 \times 10^{-6}$ for noiseless and noisy signals, respectively. Algorithms 4.1 and 4.4 were used with $\varepsilon_c = 1 \times 10^{-5}$ while Algorithm 4.2 was used with $\varepsilon_c = 1 \times 10^{-9}$. The results for noiseless signals of 20 dB are described next.

Convergence assessment

Convergence to a point in the solution set in Definition 4.1 requires the prior selection of several parameters. On the basis of Theorem 4.2, parameter selection boils down to finding an appropriate initial point and regularization and prox-parameter sequences. We employed the initial point

$$\boldsymbol{x}_{\epsilon_0}^{(0)} = \boldsymbol{A}^T \boldsymbol{b} \tag{4.146}$$

and a regularization sequence $\{\epsilon_j\}_{j\in\{0\}\cup\mathcal{I}_q}$ where the first and last terms are given by

$$\epsilon_0 = \left\lceil \log \left\| \boldsymbol{x}_{\epsilon_0}^{(0)} \right\|_\infty \right\rceil \qquad \text{and} \qquad \epsilon_q = 10^{-9} \tag{4.147}$$

with $q = 15$, respectively. The remaining terms $\epsilon_1, \ldots, \epsilon_{q-1}$ are defined by logarithmically spaced decreasing numbers between ϵ_0 and ϵ_q. From Condition 2 of Theorem 4.2 and (4.14), (4.21), (4.45), and (4.51), the terms of the prox-parameter sequence $\{\gamma_k\}_{k\in\mathbb{N}}$ lie in the open interval

$$\alpha \frac{\epsilon_j^{1-p}}{np} < \gamma_k < \frac{\epsilon_j^{2-p}}{p|p-1|}, \qquad \text{for all } j \in 0 \cup \mathcal{I}_q \text{ and } k \in \mathbb{N} \tag{4.148}$$

Assuming that the size of the neighborhood around a solution point of each problem in the sequence $\left\{(\mathrm{LP}_{\epsilon_j,\delta})\right\}_{j\in\{0\}\cup\mathcal{I}_q}$ is unknown, we express γ_k in (4.148) as

$$\gamma_k = \zeta \frac{\epsilon_j^{2-p}}{p|p-1|}, \qquad \text{for some } \zeta \in (0,1) \tag{4.149}$$

Finding a suitable sequence $\{\gamma_k\}_{k\in\mathbb{N}}$ reduces to the problem of finding values of ζ such that convergence to a solution is achieved for the value of α under consideration.

In the recovery of noiseless signals, the set $\mathcal{K}_\delta$ in (4.2) reduces to a polytope of the form

$$\mathcal{K}_\delta = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$$

and the solution set $\mathcal{S}_{\epsilon,\delta}$ in (4.23) now corresponds to the sparse limit points of the iteration sequence in (4.63) or (4.145) with at most m nonzero coordinates (see Theorem 3 of [47] and Sec. 13.2 of [85]). Thus, if we let $\bar{x}_{\epsilon_q}$ denote a limit point of (4.63) or (4.145) and let r_{ϵ_q} denote the number of nonzero coordinates of $\bar{x}_{\epsilon_q}$, then convergence to a solution can be verified by using the following criterion

$$\bar{x}_{\epsilon_q} \in \mathcal{S}_{\epsilon_q,\delta} \qquad \Longleftrightarrow \qquad r_{\epsilon_q} \leq m \tag{4.150}$$

On the basis of (4.146), (4.147), (4.149), and (4.150), we carried out simulations to estimate the probability of convergence to a solution, denoted as $\mathrm{P}\left(\bar{x}_{\epsilon_q} \in \mathcal{S}_{\epsilon_q,\delta}\right)$. The results for several values of ζ are shown in Fig. 4.1. As can be seen, values of ζ

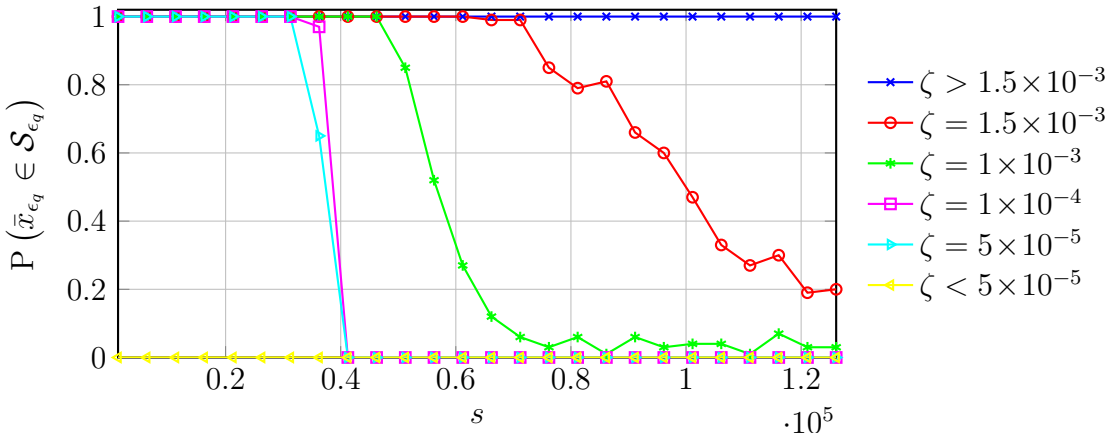


Figure 4.1: Probability of convergence in terms of ζ.

in the open interval $(0.0015, 1)$ are appropriate for signal recovery as convergence to a solution is always achieved. On the other hand, values of ζ in the open interval $(0, 0.0015)$ lead to unreliable signal recovery because the proposed method does not always converge to a solution. On the basis of Theorem 4.2, convergence is not achieved because the inequality on the left-hand side of (4.148) does not hold true for these values of ζ.

SPF evaluation

We carried out recovery simulations to evaluate the relative merits of the proposed method when different values of l in (4.77) are employed. The RP for different values of l is compared in Fig. 4.2. As can be seen, smaller values of l lead to superior RP.

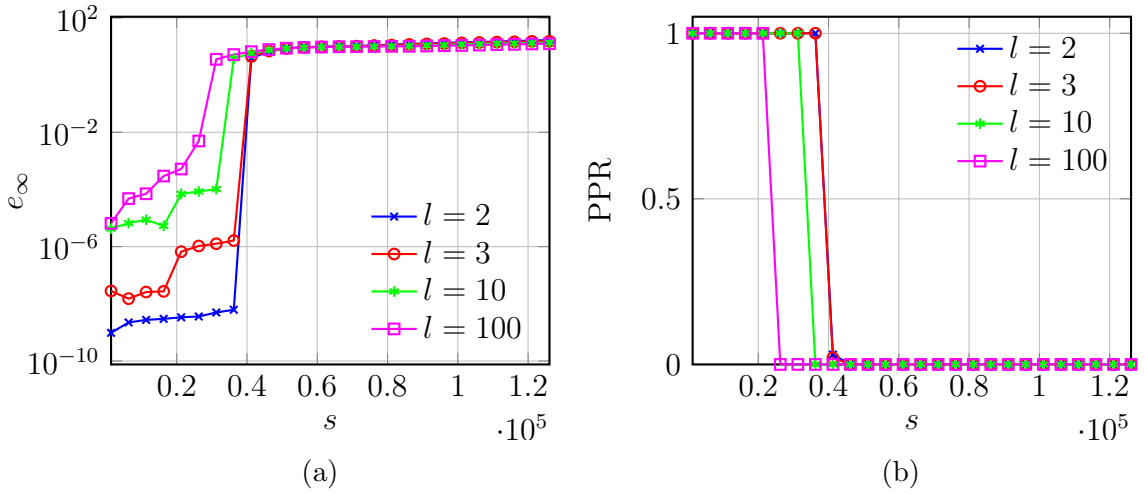


Figure 4.2: RP metrics for several values of l: (a) ℓ_∞ recovery error and (b) PPR.

For example, the average ℓ_∞ reconstruction error for $l = 2$, 3, 10, and 100 in Fig. 4.2a was 3.39×10^{-9}, 5.9×10^{-7}, 0.4783, and 1.0995, respectively, for $s \leq 36,248$. The MRF for perfect reconstruction in Fig. 4.2b has dropped from 4.2 and 6.2 when $l = 10$ and 100, respectively, to 3.6 when $l = 2$ or 3.

The CC for different values of l is compared in Fig. 4.3. As can be seen, smaller values of l lead to reduced CC in simulations where the sparse signal is always perfectly reconstructed. Conversely, the CC is not always reduced when perfect reconstruction is not always achieved. For example, for $s \leq 21,275$ the average CPU time for $l = 2$, 3, 10, and 100 in Fig. 4.3a was 323.9, 667.4, 963.5, and 856.0 seconds, respectively, and the average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ in Fig. 4.3b was 1.12×10^3, 1.3×10^3, 1.43×10^3, and 1.75×10^3, respectively. Similarly, for $s > 21,275$ the average CPU time for $l = 2$, 3, 10, and 100 in Fig. 4.3a was 1.2798×10^3, 1.7186×10^3, 2.4368×10^3, and 1.8321×10^3 seconds, respectively, and the average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ in Fig. 4.3b was 4.06×10^3, 3.39×10^3, 3.34×10^3, and 2.38×10^3, respectively.

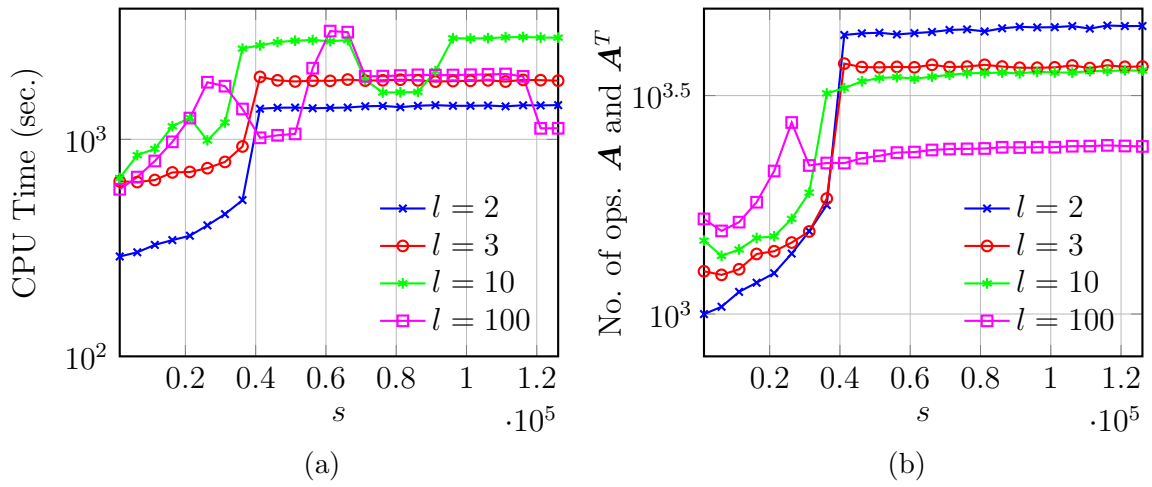


Figure 4.3: CC metrics for several values of l: (a) Average CPU time and (b) Number of matrix-vector operations with $\boldsymbol{A}$ and $\boldsymbol{A}^T$.

Two-step acceleration

We carried out simulations to evaluate the rate of convergence of the proposed IPPP and FIPPP methods. The average number of points k computed for sequence $\{\boldsymbol{x}_{\epsilon_j}^{(k)}\}_{k\in\mathbb{N}}$ to converge per continuation step j is given in Fig. 4.4. As can be seen, the two-step acceleration leads to fast convergence. For example, the average number of iterations required for perfect reconstruction of sparse signals with $s = 1,311$ and $s = 21,275$ in Figs. 4.4a and 4.4b has dropped from 465.4 and 818.1 with the IPPP method to 287.2 and 346.6 with FIPPP method, respectively, This corresponds to a 38% and 58% decrease in the average number of iterations. Faster convergence is also achieved when the sparse signal is not always perfectly reconstructed. For example, the average number of iterations for $s = 96,140$ and $s = 121,095$ in Figs. 4.4c and 4.4d has dropped from 3.3828×10^3 and 3.3997×10^3 with the IPPP method to 1.2714×10^3 and 1.2744×10^3 with the FIPPP method, respectively. This corresponds to a 62% and 63% decrease in the average number of iterations.

4.4.2 Comparison of the proposed BP method with state-of-the-art competing methods

We carried out recovery simulations to evaluate the proposed and competing methods for the recovery of noiseless and noisy signals. Results obtained for noiseless and noisy signals with sparsity of $s \leq 36,248$ are summarized in Tables 4.1 and 4.2, respectively.

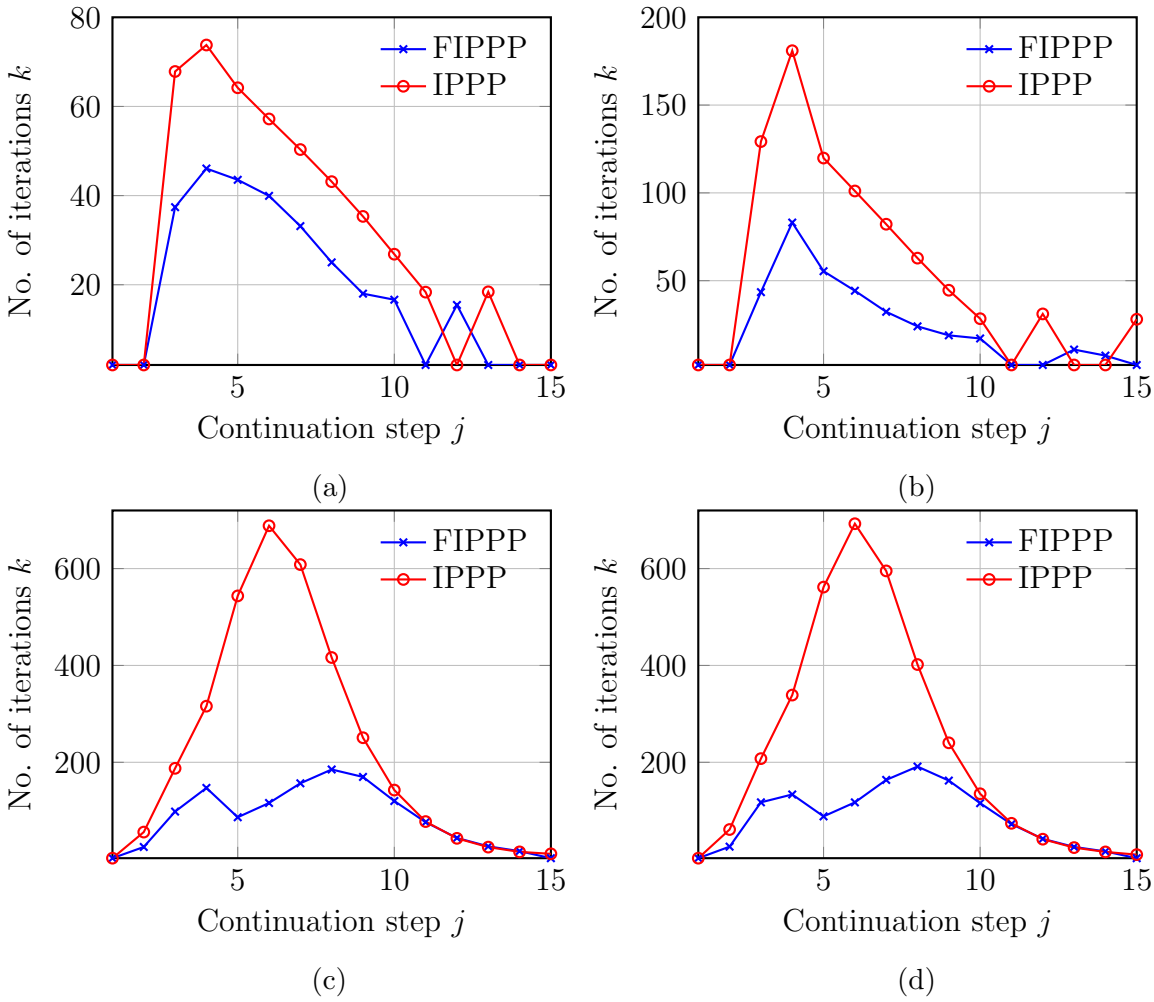


Figure 4.4: Number of iterations of the IPPP and FIPPP methods per continuation step, (a) $s = 1,311$, (b) $s = 21,275$, (c) $s = 96,140$, and (d) $s = 121,095$.

As can be seen, the FIPPP method achieved superior RP and MC and comparable CC metrics relative to those of the SPGL1 and NESTA methods. The percent change in the metrics of the competing methods compared to the metrics of the proposed method is given in Table 4.3. As can be seen, the number of measurements required by the FIPPP method to represent signals is significantly reduced relative to those of the competing methods. We observe a decrease between 41% and 86% in the MRF. On the other hand, the average CPU time required by the FIPPP method to reconstruct signals is comparable to those of the competing methods. For example, a decrease of at most 60% in the average CPU time is achieved for signals of 20 dB while an increase of at most 75% is achieved for signals of 40 dB. The results summarized in Tables 4.1 to 4.3 are described in detail in the following subsections.

Results for noiseless signals

The RP of the FIPPP method is compared with those of the SPGL1 and NESTA methods in Figs. 4.5 and 4.6. As can be seen, the proposed method achieved superior RP relative to the other methods for a variety of dynamic ranges. For example, the average ℓ_∞ reconstruction error of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.5a were 4.101×10^{-9}, 1.6546, and 1.6393, respectively, for $s \leq 36,248$. The MRF for perfect reconstruction in Fig. 4.6a has dropped from 8.0 and 6.2 with the SPGL1 and NESTA methods to 3.6 with the FIPPP method corresponding to a 55% and 41% decrease in the MRF, respectively. Similarly, the average ℓ_∞ reconstruction error of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.5d were 7.7161×10^{-6}, 18.0536, and 19.7608, respectively, for $s \leq 36,248$. The MRF for perfect reconstruction in Fig. 4.6d has dropped from 8.0 and 6.1 with the SPGL1 and NESTA methods to 1.7 with the FIPPP method corresponding to a 79% and 72% decrease in the MRF, respectively.

The CC of the FIPPP method is compared with those of the SPGL1 and NESTA

	RP		CC		MC		
DR	e_∞	MRF	CPU time	no. ops. $\boldsymbol{A}$ & $\boldsymbol{A}^T$	median of $\|\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|\|$	δ	method
	1.6546	8.0	584.3	1.12×10^3	8.01×10^{-5}		SPGL1
20	1.6393	6.2	925.3	3.37×10^3	1.08×10^{-13}	0	NESTA
	4.1×10^{-9}	3.6	372.6	1.29×10^3	1.16×10^{-13}		FIPPP
	3.0398	11.6	752.2	1.26×10^3	7.35×10^{-5}		SPGL1
40	4.3709	8.0	221.2	877.6	8.47×10^{-13}	0	NESTA
	1.3×10^{-8}	2.6	359.1	1.22×10^3	8.20×10^{-13}		FIPPP
	9.9265	11.6	802.4	1.48×10^3	7.55×10^{-5}		SPGL1
80	14.2567	6.2	381.1	1.57×10^3	6.39×10^{-11}	0	NESTA
	3×10^{-7}	1.8	388.1	1.34×10^3	5.96×10^{-11}		FIPPP
	18.0536	8.0	873.5	1.56×10^3	8.29×10^{-5}		SPGL1
100	19.7608	6.2	565.3	2.25×10^3	5.75×10^{-10}	0	NESTA
	7.7×10^{-6}	1.7	394.5	1.36×10^3	5.34×10^{-10}		FIPPP

Table 4.1: Summary of results for noiseless signals.

methods in Figs. 4.7 and 4.8. As can be seen, the proposed method achieved comparable CC relative to the other methods for a variety of dynamic ranges in simulations where the sparse signal is always perfectly reconstructed. For example, the average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.7a were 372.6, 584.3, and 925.3 seconds for $s \leq 36,248$, respectively, corresponding to a 36% and 60% decrease in the average CPU time obtained with the SPGL1 and NESTA methods. The average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.8a were 1.2941×10^3, 1.1257×10^3, and 3.375×10^3 for $s \leq 36,248$, respectively, corresponding to a 15% increase and 62% decrease in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods. Similarly, the average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.7d were 394.5, 873.5, and 565.3 seconds for $s \leq 36,248$, respectively, corresponding to a 55% and 30% decrease in the average CPU time obtained with the SPGL1 and NESTA methods. The average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of

	RP		CC		MC		
DR	e_∞	MRF	CPU time	no. ops. $\boldsymbol{A}$ & $\boldsymbol{A}^T$	median of $\|\|\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}\|\|$	δ	method
	1.6528	8.0	313.4	585.5	0.036271		SPGL1
20	1.6508	6.2	782.5	3.26×10^3	0.036204	0.036204	NESTA
	0.0017	3.6	436.8	1.47×10^3	0.0362		FIPPP
	3.0419	8.0	360.2	673.6	0.036254		SPGL1
40	4.3735	8.0	235.8	887.2	0.036204	0.036204	NESTA
	0.0018	2.6	411.8	1.38×10^3	0.036204		FIPPP
	9.8627	8.0	435.5	865.6	0.036262		SPGL1
80	14.2652	6.2	388.7	1.56×10^3	0.036203	0.036204	NESTA
	0.0020	1.8	390.1	1.41×10^3	0.036204		FIPPP
	17.9831	8.0	499.3	989.3	0.036261		SPGL1
100	19.7632	6.2	560.7	2.24×10^3	0.036202	0.036204	NESTA
	0.0021	1.7	390.1	1.40×10^3	0.036204		FIPPP

Table 4.2: Summary of results for noisy signals.

method	metrics of FIPPP			DR
	MRF	CPU time	no. ops. $\boldsymbol{A}$ & $\boldsymbol{A}^T$	
SPGL1	−55%	−36%	+16%	20
	−77%	−52%	−03%	40
	−86%	−52%	−09%	80
	−79%	−55%	−13%	100
NESTA	−41%	−60%	−62%	20
	−68%	+62%	+39%	40
	−73%	+02%	−15%	80
	−72%	−30%	−40%	100

(a)

method	metrics of FIPPP			DR
	MRF	CPU time	no. ops. $\boldsymbol{A}$ & $\boldsymbol{A}^T$	
SPGL1	−55%	+39%	+151%	20
	−68%	+14%	+105%	40
	−77%	−10%	+63%	80
	−79%	−22%	+42%	100
NESTA	−41%	−44%	−55%	20
	−68%	+75%	+56%	40
	−70%	+01%	−10%	80
	−72%	−30%	−37%	100

(b)

Table 4.3: Percent change in performance metrics of the proposed method compared to competing methods, (a) noiseless and (b) noisy signals.

20 dB in Fig. 4.8d were 1.3616×10^3, 1.5616×10^3, and 2.2489×10^3 for $s \leq 36,248$, respectively, corresponding to a 13% and 39% decrease in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods.

The CC of the proposed method is increased relative to that of the NESTA method and similar to that of the SPGL1 method for simulations where the sparse signal was not always perfectly recovered. For example, the average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.7a were 1.6306×10^3, 1.1224×10^3, and 297.7 seconds for $s > 36,248$, respectively, corresponding to a 45% and 448% increase in the average CPU time obtained with the SPGL1 and NESTA methods. The average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.8a were 5.5609×10^3, 2.292×10^3, and 978.3 for $s > 36,248$, respectively, corresponding to a 143% and 468% increase in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods. Similarly, the average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.7d were 1.1311×10^3, 2.514×10^3, and 545.1 seconds for $s > 36,248$, respectively, corresponding to a 55% decrease and 108% increase in the average CPU time obtained with the SPGL1 and NESTA methods. The average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of 100

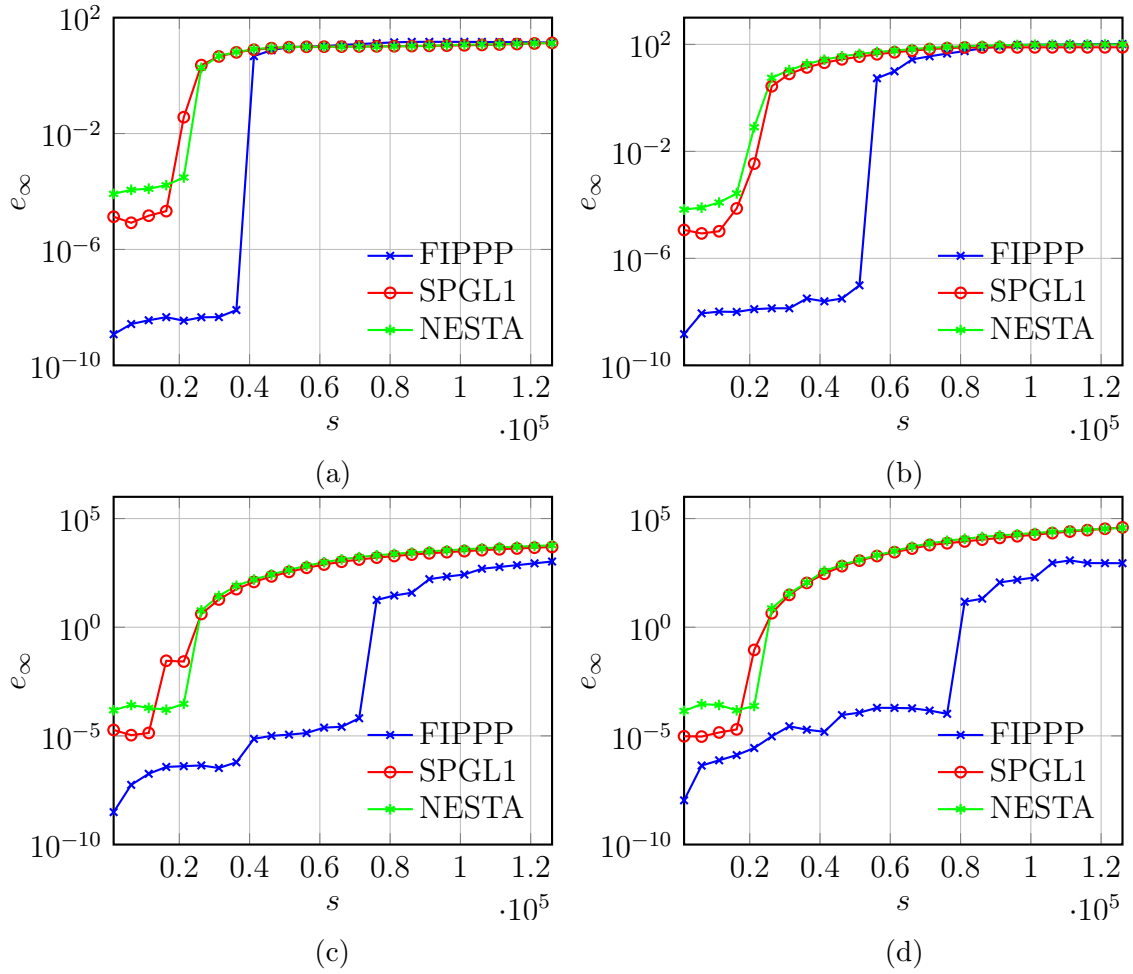


Figure 4.5: Average ℓ_∞ recovery error of the FIPPP and competing methods for noiseless signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

dB in Fig. 4.8d were 3.7872×10^3, 4.4293×10^3, and 2.175×10^3 for $s > 36,248$, respectively, corresponding to a 14% decrease and 74% increase in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods.

The MC of signals recovered by the FIPPP method is compared with those of the SPGL1 and NESTA methods in Figs. 4.9 to 4.12. Here MC is measured in terms of how close $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ is to the target value of $\delta = 0$ (see Fig. 1.4). As can be seen in the box plots[2] of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$, signals recovered with the FIPPP method are more consistent with measurements taken than those recovered with the SPGL1 and NESTA methods for a variety of dynamic ranges. For example, for simulations with signals of 20 dB and $s = 36,248$, the median and the minimum and maximum

[2]A brief explanation of box plots can be found on p. 19. See [77] for a detailed description.

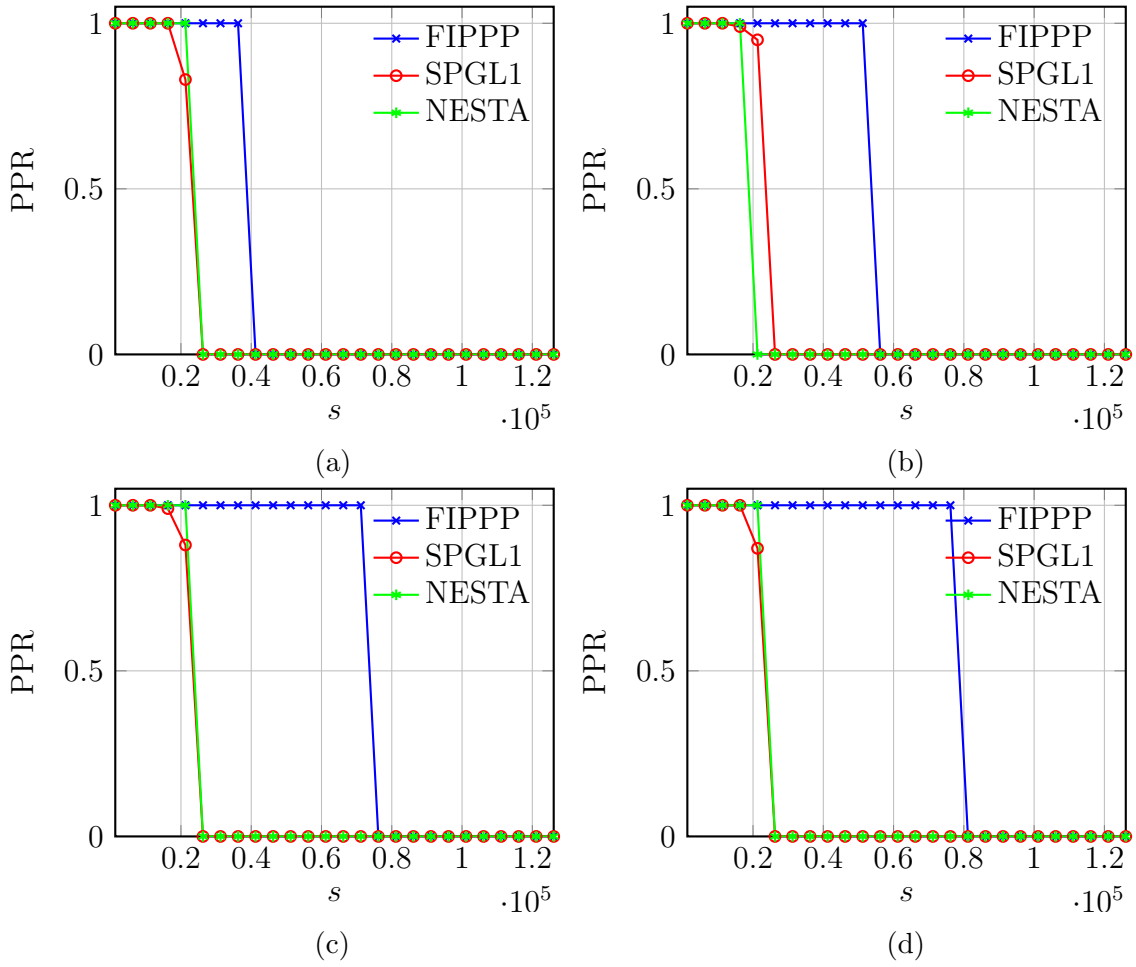


Figure 4.6: PPR of the FIPPP and competing methods for noiseless signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the FIPPP method were 1.1572×10^{-13}, 1.1479×10^{-13}, and 1.1682×10^{-13}, respectively, as shown in Fig. 4.9a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the SPGL1 method were 8.0102×10^{-5}, 2.9597×10^{-5}, and 9.9963×10^{-5} as shown in Fig. 4.9b while those obtained with the NESTA method were 1.0757×10^{-13}, 1.0651×10^{-13}, and 1.0851×10^{-13} as shown in Fig. 4.9c. For simulations with signals of 100 dB and $s = 36,248$, the median and the minimum and maximum observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the FIPPP method were 5.3438×10^{-10}, 5.2018×10^{-10}, and 5.4575×10^{-10}, respectively, as shown in Fig. 4.12a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the SPGL1 method were 8.2889×10^{-5}, 3.1279×10^{-5}, and 9.9983×10^{-5} as shown in Fig. 4.12b while those obtained with the NESTA method were 5.7474×10^{-10}, 5.5964×10^{-10},

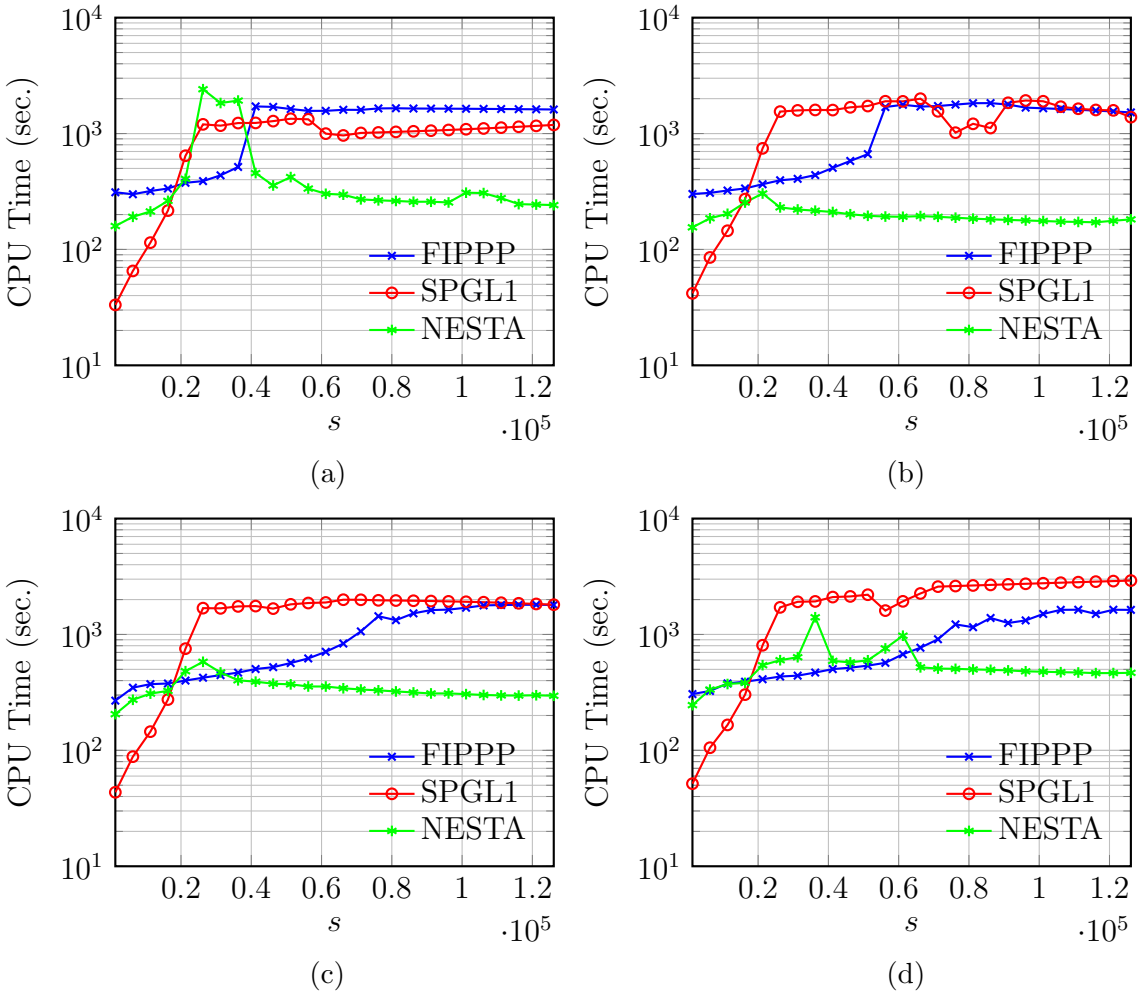


Figure 4.7: Average CPU time of the FIPPP and competing methods for noiseless signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

and 5.8649×10^{-10} as shown in Fig. 4.12c.

Results for noisy signals

The RP of the FIPPP method is compared with those of the SPGL1 and NESTA methods in Figs. 4.13 and 4.14. As can be seen, the proposed method achieved superior RP relative to the other methods for a variety of dynamic ranges. For example, the average ℓ_∞ reconstruction error of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.13a were 0.0017, 1.6528, and 1.6508, respectively, for $s \leq 36,248$. The MRF for perfect reconstruction in Fig. 4.14a has dropped from 8.0 and 6.2 with the SPGL1 and NESTA methods to 3.6 with the FIPPP method corre-

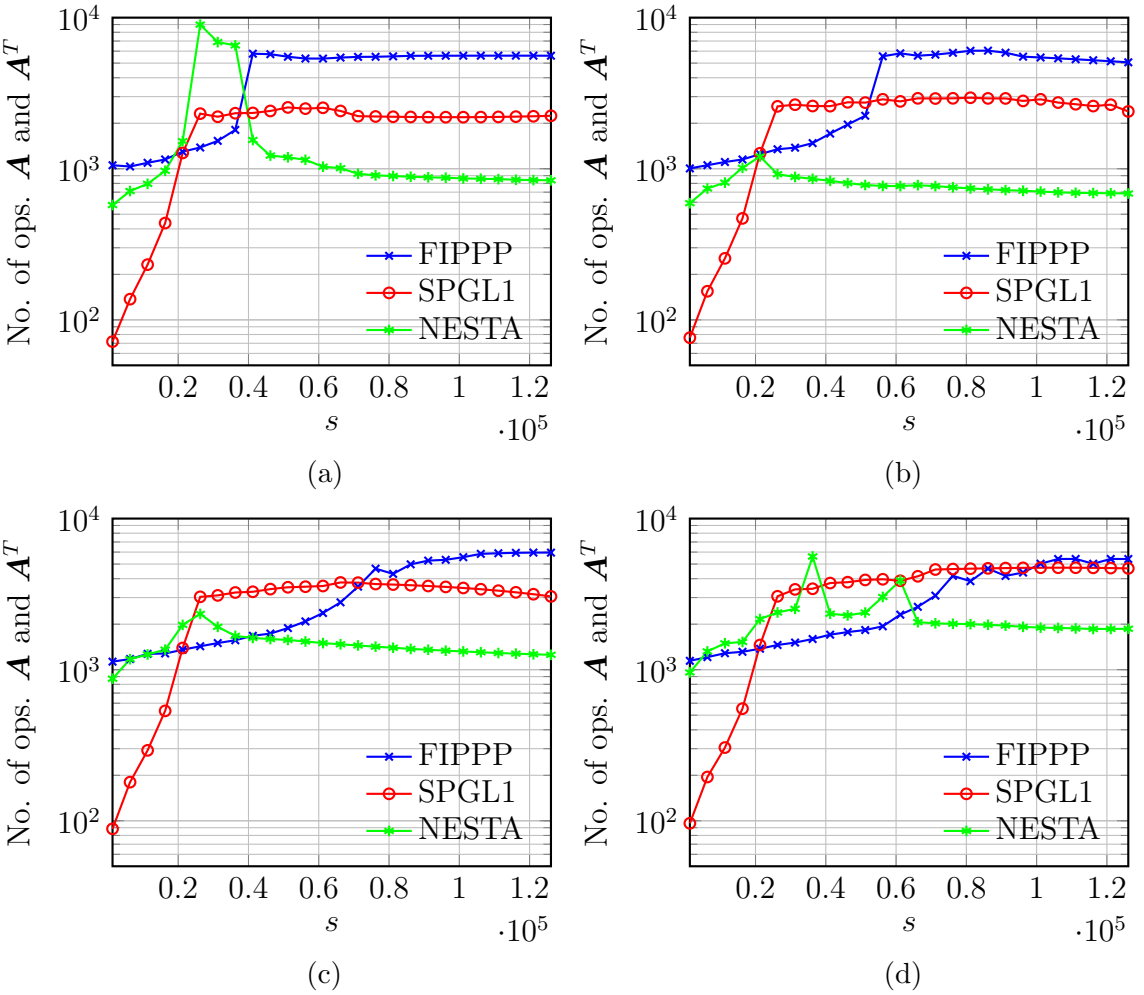


Figure 4.8: Number of matrix-vector operations with $\boldsymbol{A}$ and $\boldsymbol{A}^T$ for the FIPPP and competing methods for noiseless signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

sponding to a 55% and 41% decrease in the MRF, respectively. Similarly, the average ℓ_∞ reconstruction error of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.13d were 0.0021, 17.9831, and 19.7632, respectively, for $s \leq 36,248$. The MRF for perfect reconstruction in Fig. 4.14d has dropped from 8.0 and 6.2 with the SPGL1 and NESTA methods to 1.7 with the FIPPP method corresponding to a 79% and 72% decrease in the MRF, respectively.

The CC of the FIPPP method is compared with those of the SPGL1 and NESTA methods in Figs. 4.15 and 4.16. As can be seen, the proposed method achieved comparable CC relative to the other methods for a variety of dynamic ranges in simulations where the sparse signal is always perfectly reconstructed. For example, the

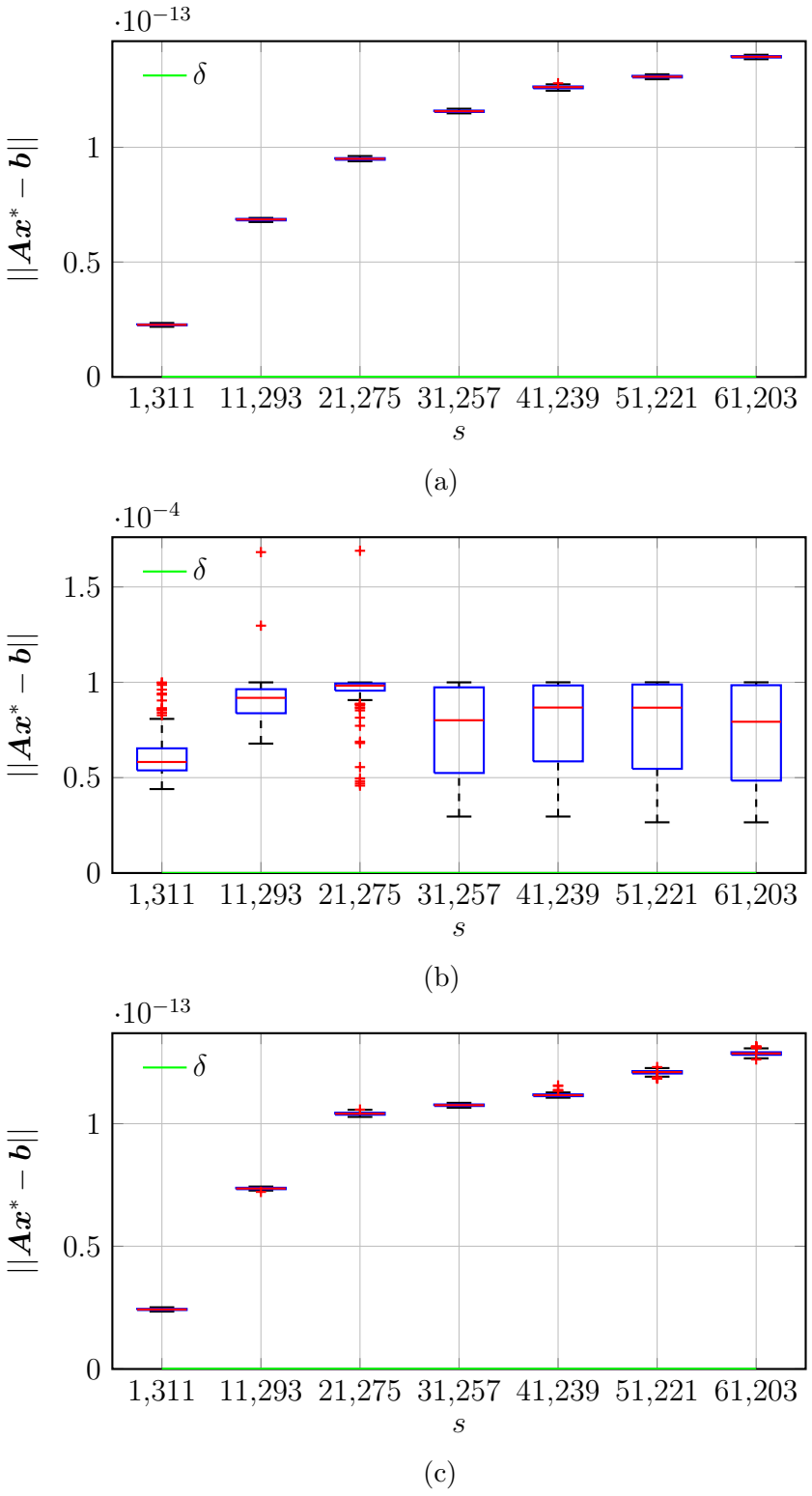


Figure 4.9: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noiseless signals of 20 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.

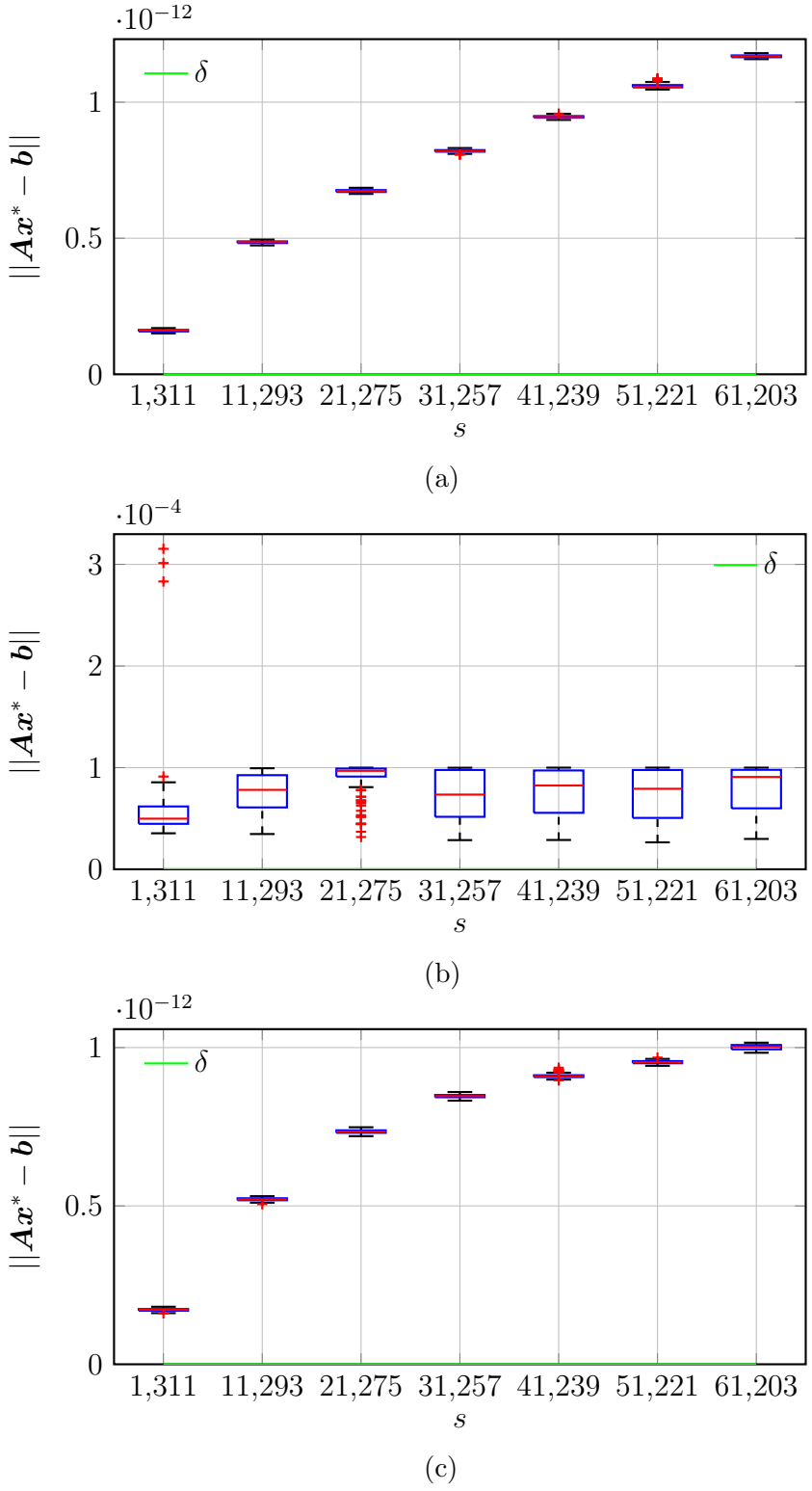


Figure 4.10: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noiseless signals of 40 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.

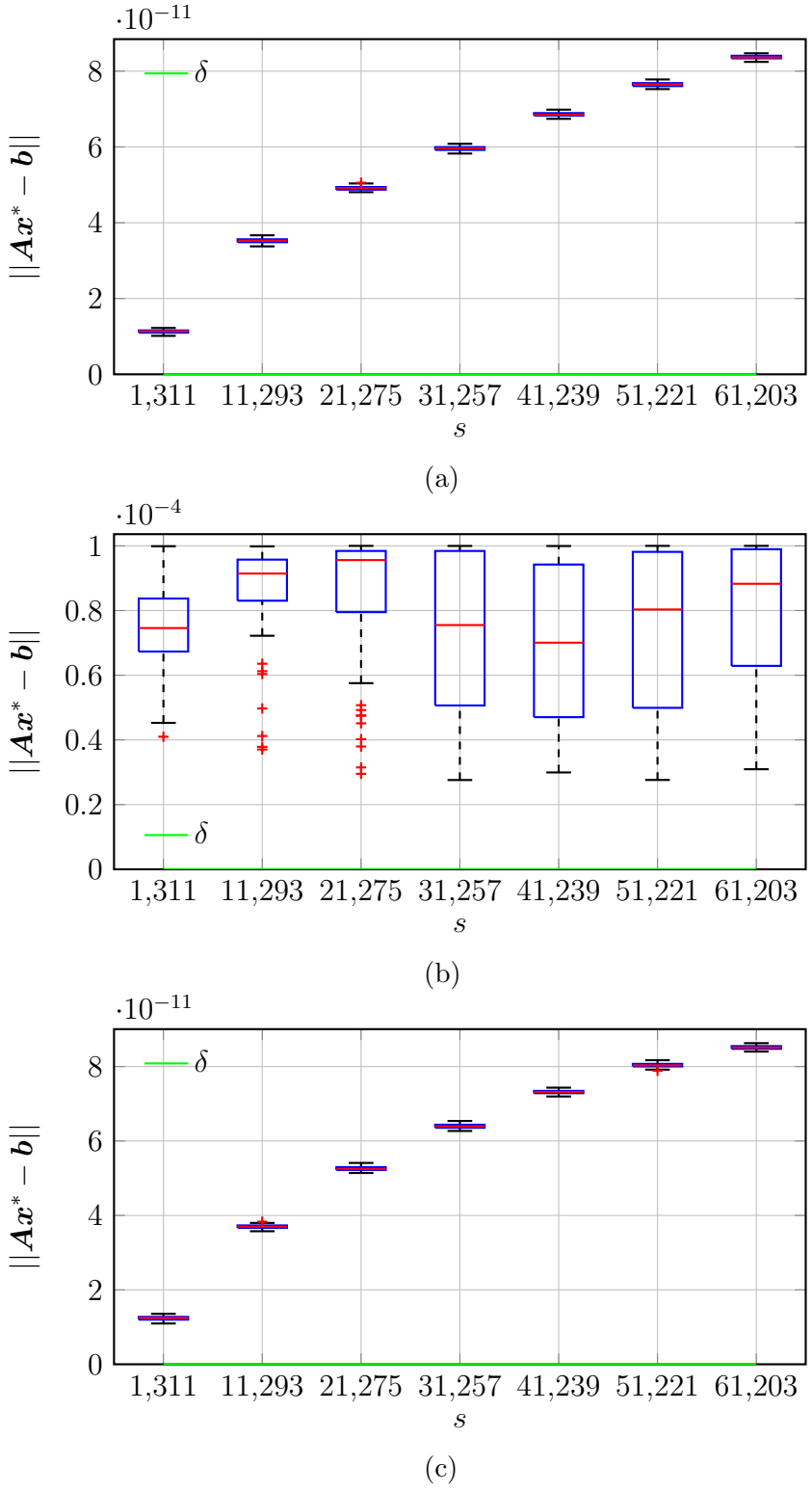


Figure 4.11: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noiseless signals of 80 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.

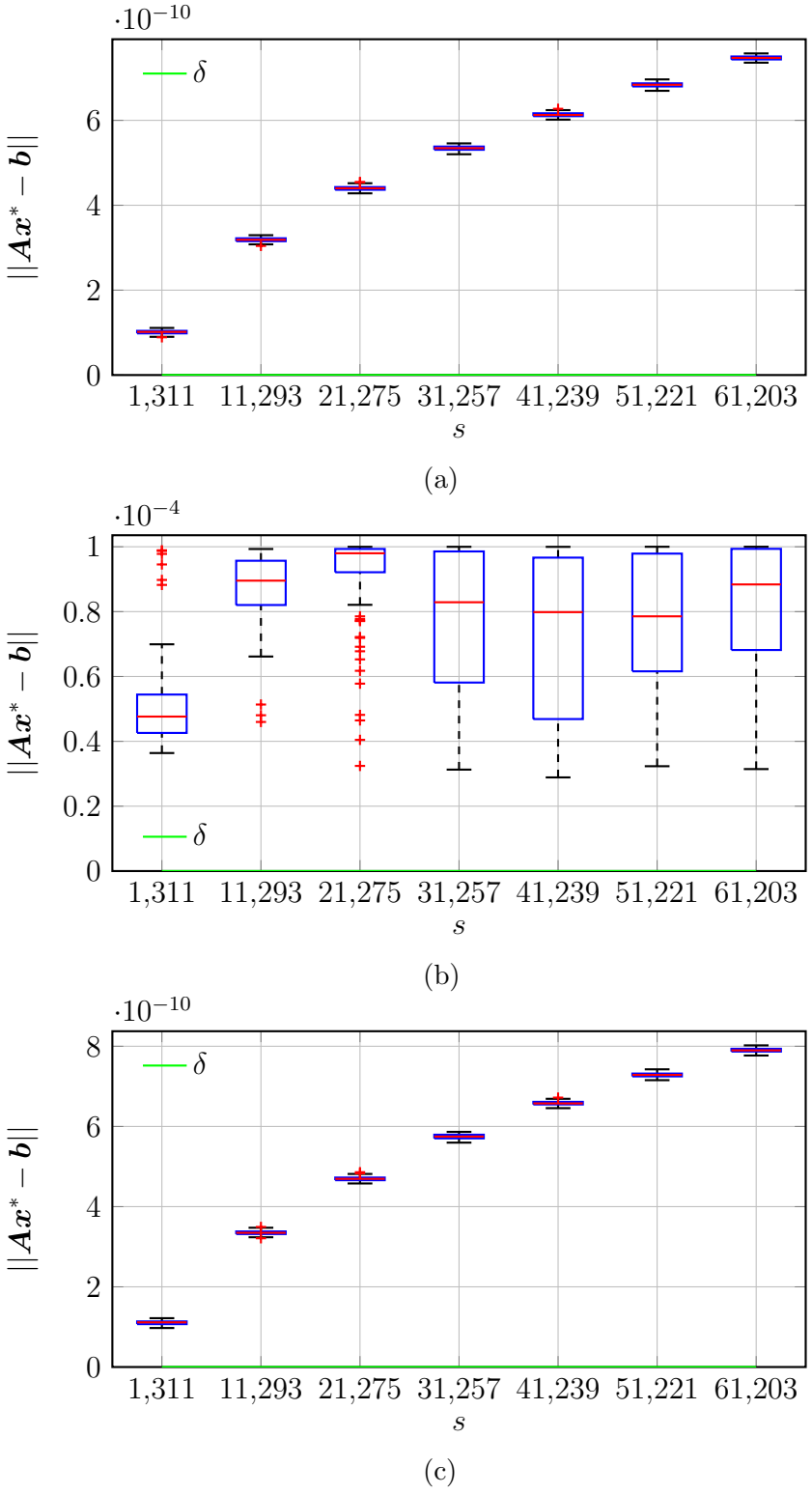


Figure 4.12: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noiseless signals of 100 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.

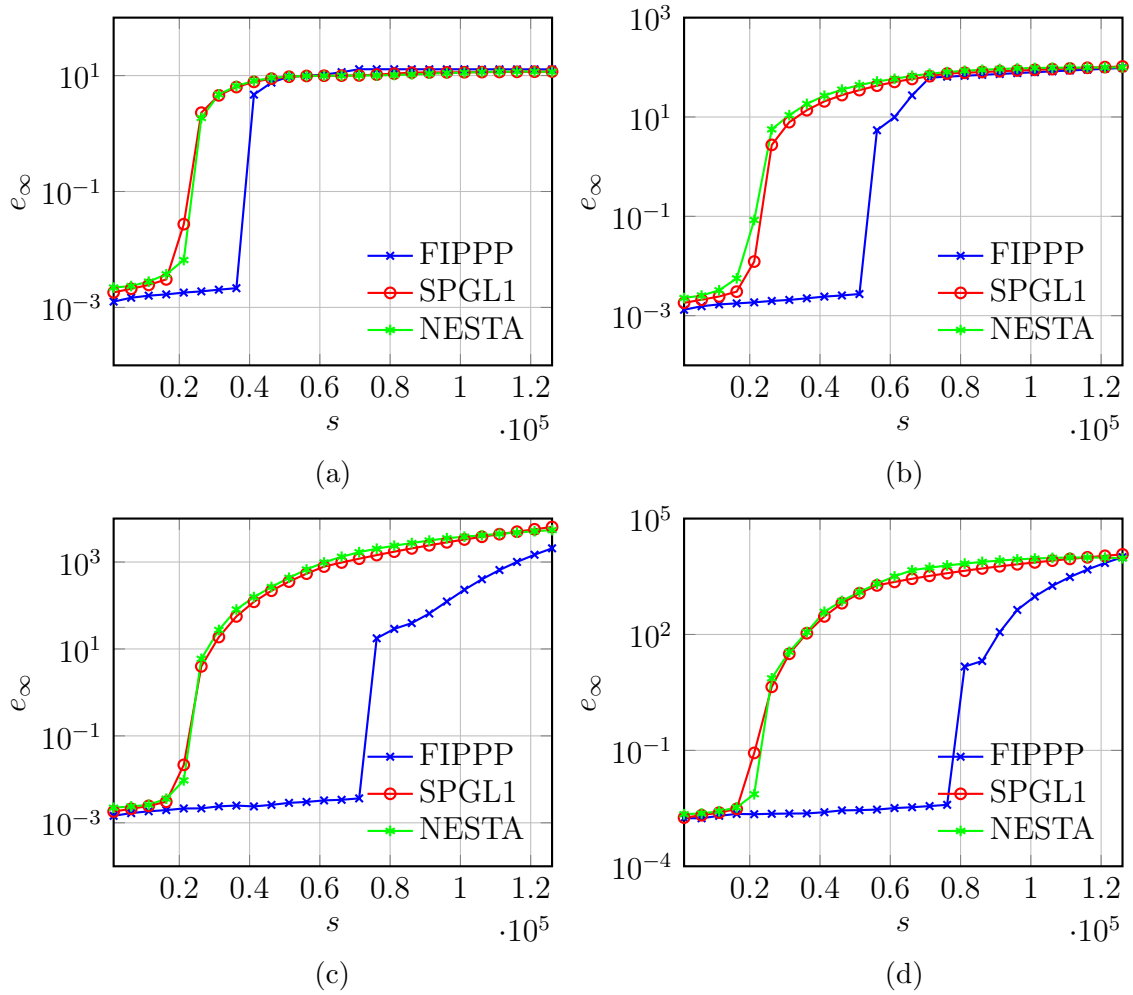


Figure 4.13: Average ℓ_∞ recovery error of the FIPPP and competing methods for noisy signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.15a were 436.8, 313.5, and 782.5 seconds for $s \leq 36,248$, respectively, corresponding to a 39% increase and 44% decrease in the average CPU time obtained with the SPGL1 and NESTA methods. The average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.16a were 1.4676×10^3, 585.5, and 3.256×10^3 for $s \leq 36,248$, respectively, corresponding to a 151% increase and 55% decrease in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods. Similarly, the average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.15d were 390.1, 499.3, and 560.7 seconds for $s \leq 36,248$, respectively, corresponding to a 22% and 30% decrease in the average CPU time obtained with

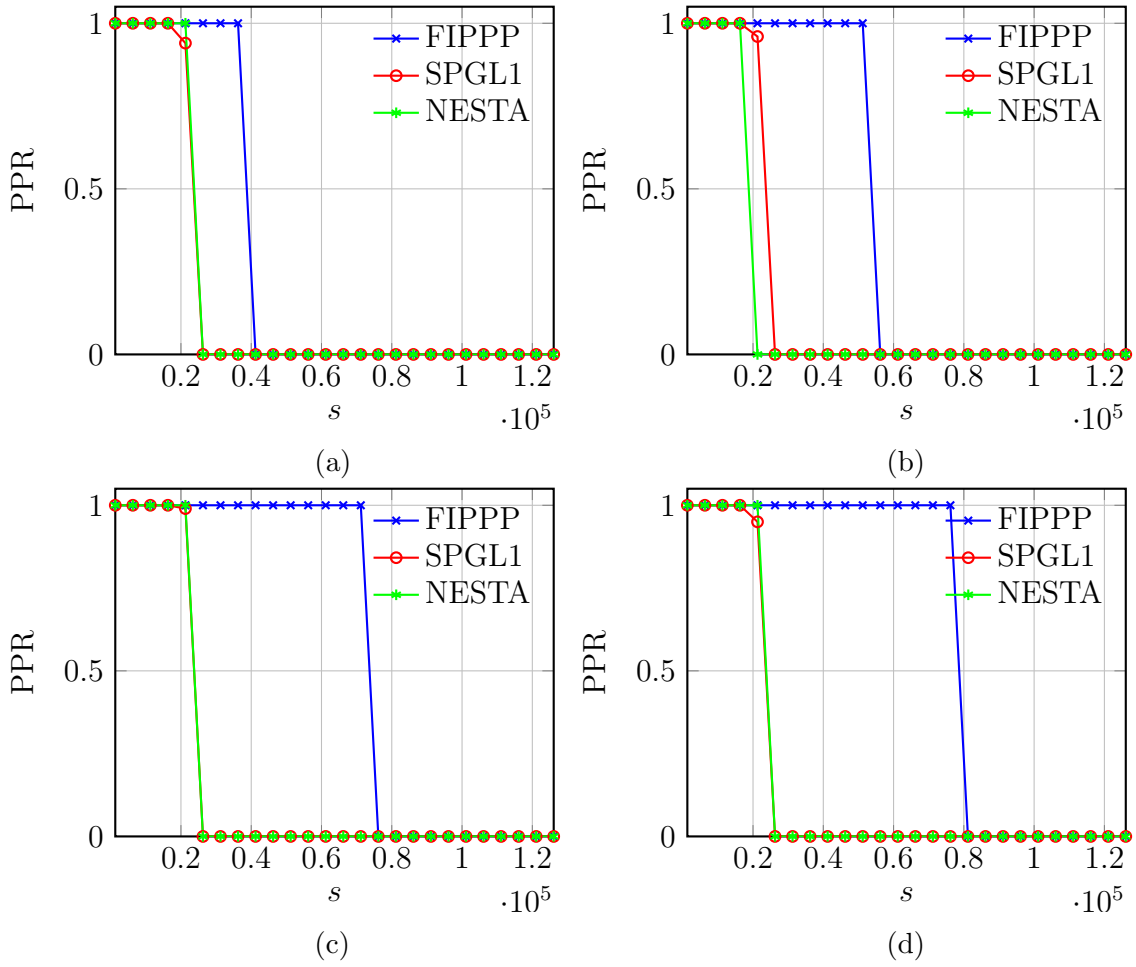


Figure 4.14: PPR of the FIPPP and competing methods for noisy signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

the SPGL1 and NESTA methods. The average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.16d were 1.4045×10^3, 989.3, and 2.2447×10^3 for $s \leq 36,248$, respectively, corresponding to a 42% increase and 37% decrease in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods.

The CC of the proposed method is increased relative to those of the competing methods for simulations where the sparse signal was not always perfectly recovered. For example, the average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.15a were 1.514×10^3, 657.3, and 132.6 seconds for $s > 36,248$, respectively, corresponding to a 130% and 1,042% increase in the average CPU time obtained with the SPGL1 and NESTA methods. The average number of

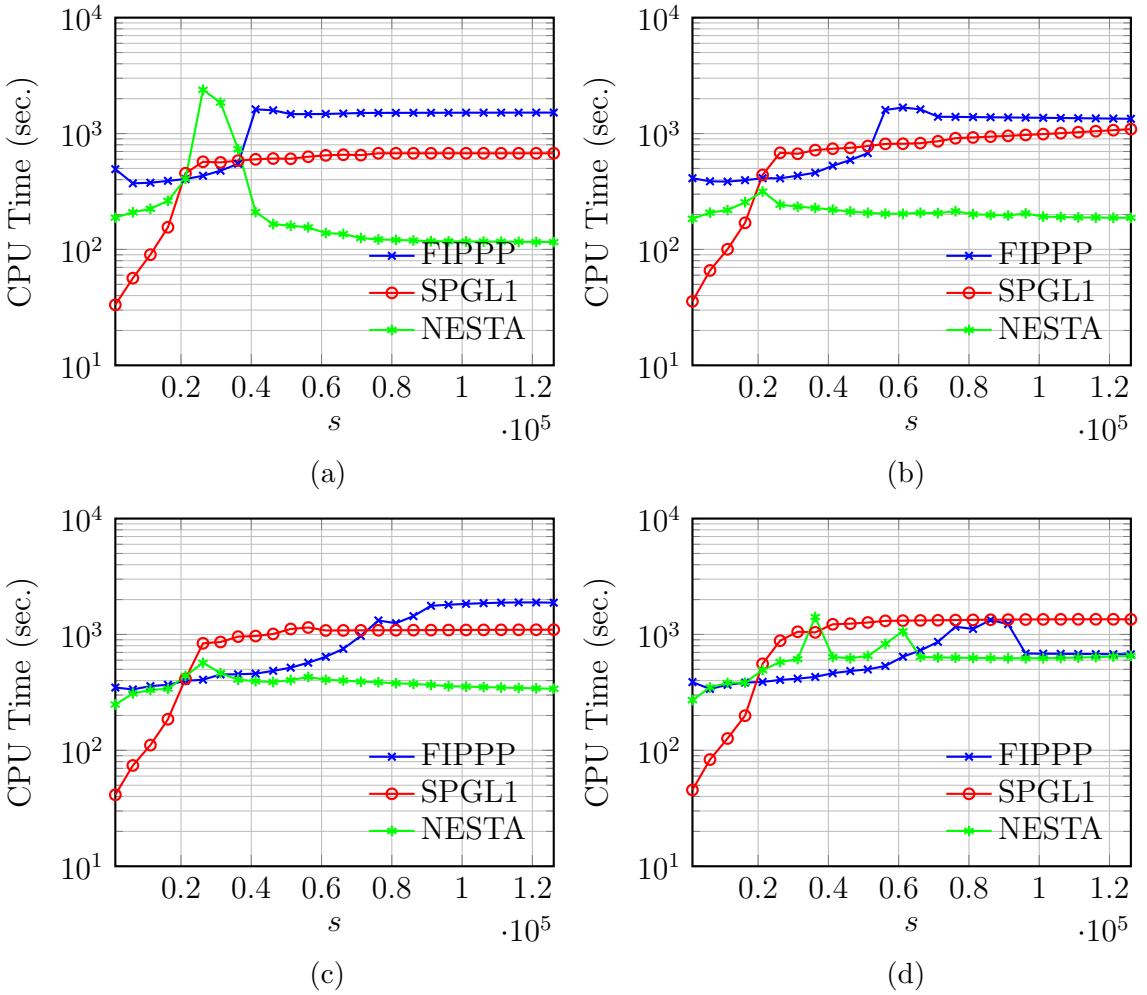


Figure 4.15: Average CPU time of the FIPPP and competing methods for noisy signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of 20 dB in Fig. 4.16a were 4.9758×10^3, 1.1503×10^3, and 965.1 for $s > 36,248$, respectively, corresponding to a 333% and 416% increase in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods. Similarly, the average CPU time of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.15d were 767.4, 1.3228×10^3, and 668.3 seconds for $s > 36,248$, respectively, corresponding to a 42% decrease and 15% increase in the average CPU time obtained with the SPGL1 and NESTA methods. The average number of matrix-vector operations with matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ of the FIPPP, SPGL1, and NESTA methods for signals of 100 dB in Fig. 4.16d were 2.5744×10^3, 2.5435×10^3, and 2.4303×10^3 for $s > 36,248$, respectively, corresponding to a 1% and 6% increase

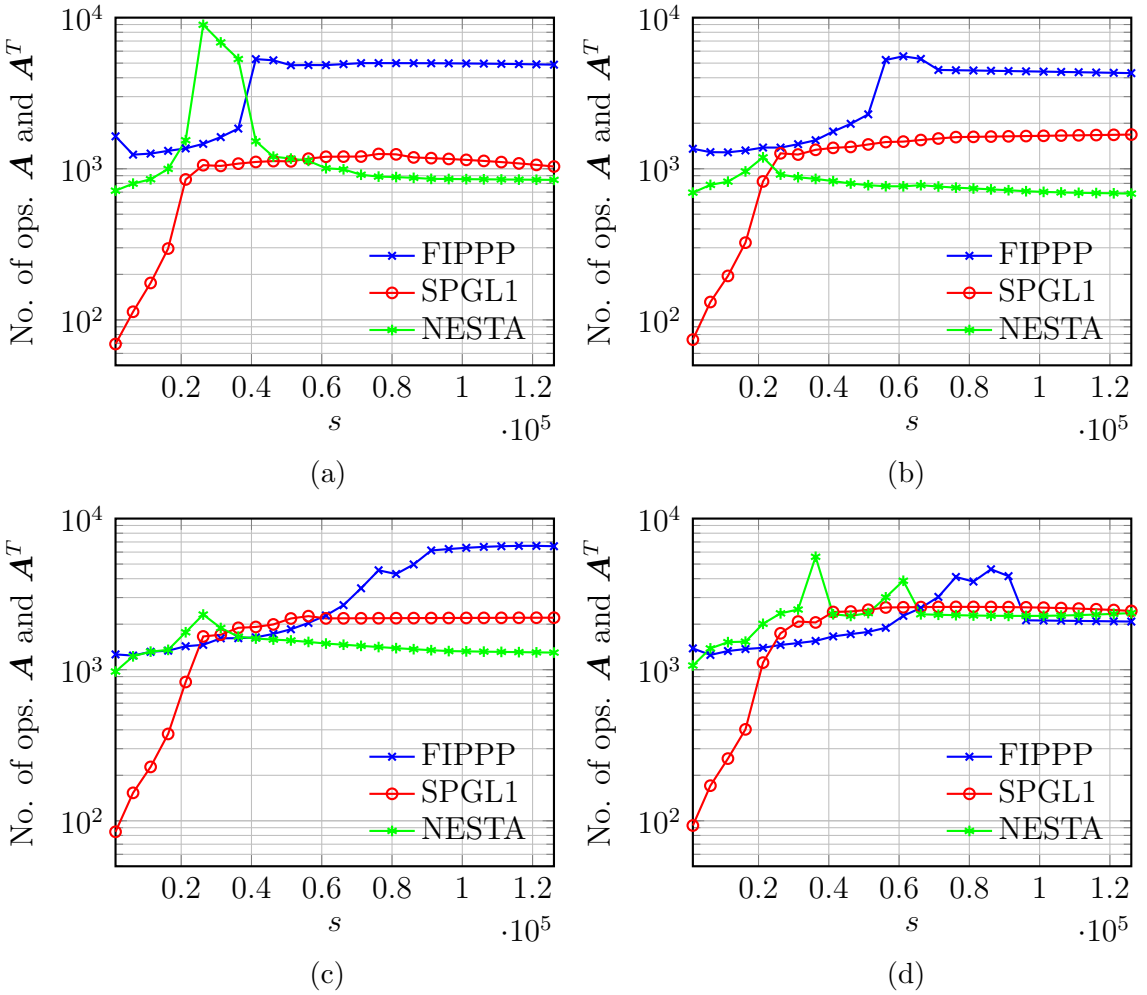


Figure 4.16: Number of matrix-vector operations with $\boldsymbol{A}$ and $\boldsymbol{A}^T$ for the FIPPP and competing methods for noisy signals, (a) 20 dB, (b) 40 dB, (c) 80 dB, and (d) 100 dB signals.

in the average number of matrix-vector operations obtained with the SPGL1 and NESTA methods.

The MC of signals recovered by the FIPPP method is compared with those of the SPGL1 and NESTA methods in Figs. 4.17 to 4.20. Here MC is measured in terms of how close $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ is to the target value of $\delta = 0.03620387$ (see Fig. 1.4). As can be seen in the box plots of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$, signals recovered with the FIPPP method are more consistent with measurements taken than those recovered with the SPGL1 and NESTA methods for a variety of dynamic ranges. For example, for simulations with signals of 20 dB and $s = 36,248$, the median and the minimum and maximum observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the FIPPP method were 0.0362, 0.0362, and

0.036204, respectively, as shown in Fig. 4.17a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the SPGL1 method were 0.036271, 0.036105, and 0.036304 as shown in Fig. 4.17b while those obtained with the NESTA method were 0.036204, 0.036204, and 0.036204 as shown in Fig. 4.17c. For simulations with signals of 100 dB and $s = 36,248$, the median and the minimum and maximum observations of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the FIPPP method were 0.036204, 0.035832, and 0.036204, respectively, as shown in Fig. 4.20a. On the other hand, the statistics of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ obtained with the SPGL1 method were 0.036261, 0.036121, and 0.036304 as shown in Fig. 4.20b while those obtained with the NESTA method were 0.036202, 0.036202, and 0.036202 as shown in Fig. 4.20c.

4.5 Conclusions

A new signal-recovery method has been described. Sparse-signal recovery is carried out by minimizing the sum of the ϵ-ℓ_p^p norm function and the indicator function of a closed ball an affine mapping. The objective function obtained in this way exhibits rich properties such as a convex and differentiable ME and a cohypomonotone subgradient mapping. A PP method is used for minimizing the objective function and a continuation procedure with a suitable regularization sequence is employed so that a minimum can be found efficiently. When the iteration sequence is computed approximately, the method is applied by iteratively performing two fundamental operations, namely, computation of the PP of the ϵ-ℓ_p^p norm function and projection of the PP onto the closed ball under affine mapping.

The first operation boils down to finding the largest real root of a trinomial equation when $p \in \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \ldots\}$ and is performed analytically or numerically by using a fast iterative method. The second operation boils down to finding a point in the intersection of convex sets and is performed efficiently by computing a sequence of closed-form projectors while using matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ in matrix-vector operations only. The error sequence entailed by the approximate computation is shown to be monotonically decreasing and summable. Consequently, the sequence of points associated with the iterative computation converges to a minimizer. Accelerated convergence is achieved by using a two-step method with optimal convergence rate.

Simulations demonstrate that very-large signals with a wide dynamic range can be recovered accurately and efficiently using the proposed method. The results obtained show that superior RP metrics such as increased PPRs, reduced MRFs for perfect

recovery, and reduced average ℓ_∞ reconstruction error are achieved when using the proposed method relative to the SPGL1 [11] and NESTA [9] methods. In addition, superior MC metrics based on the difference between $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||_2$ and δ are achieved when using the proposed method relative to the competing methods. CC metrics such as the average CPU time and number of matrix-vector operations with $\boldsymbol{A}$ and $\boldsymbol{A}^T$ required by the proposed method were found to be comparable to that of the competing methods.

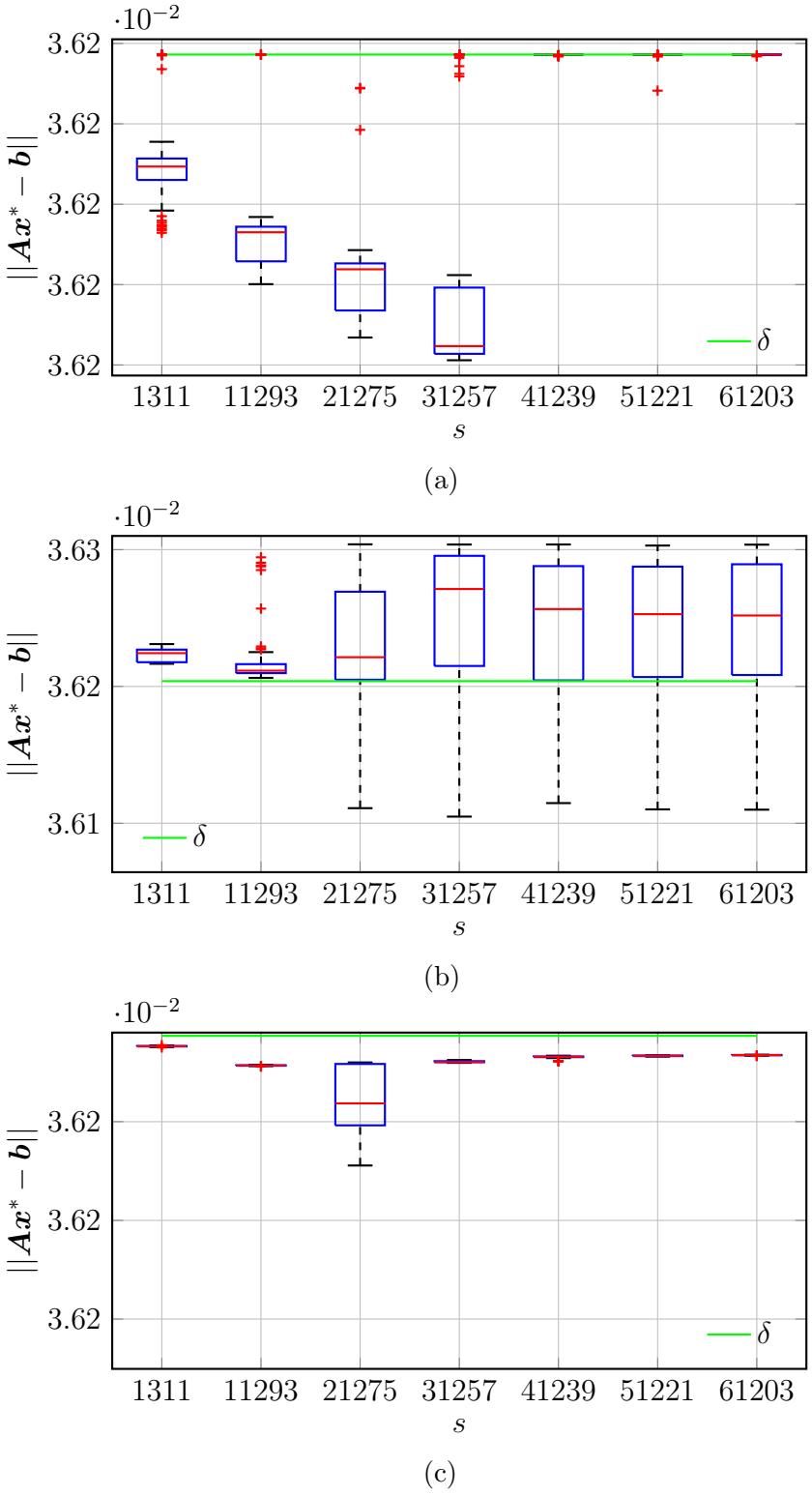


Figure 4.17: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noisy signals of 20 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.

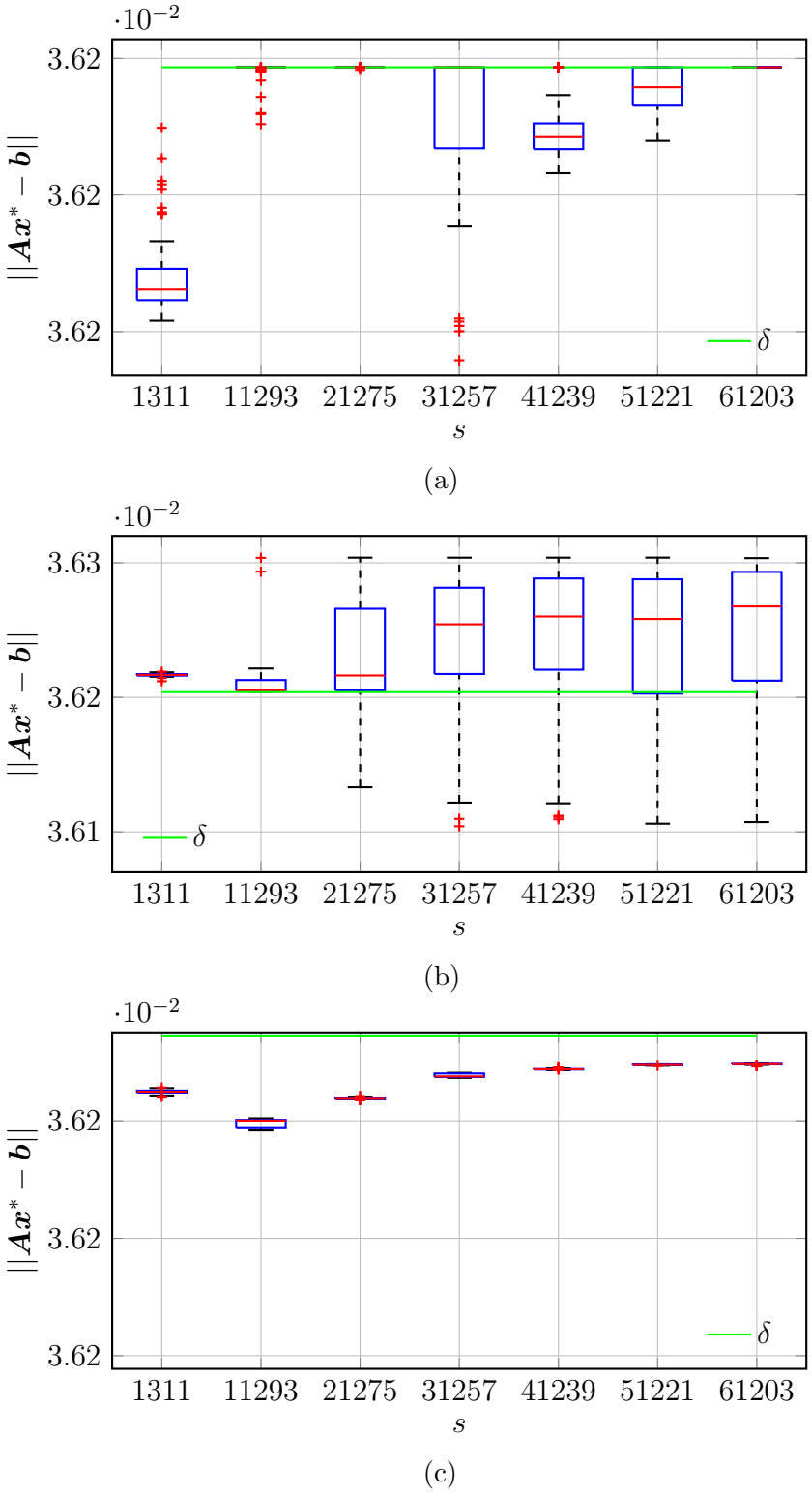
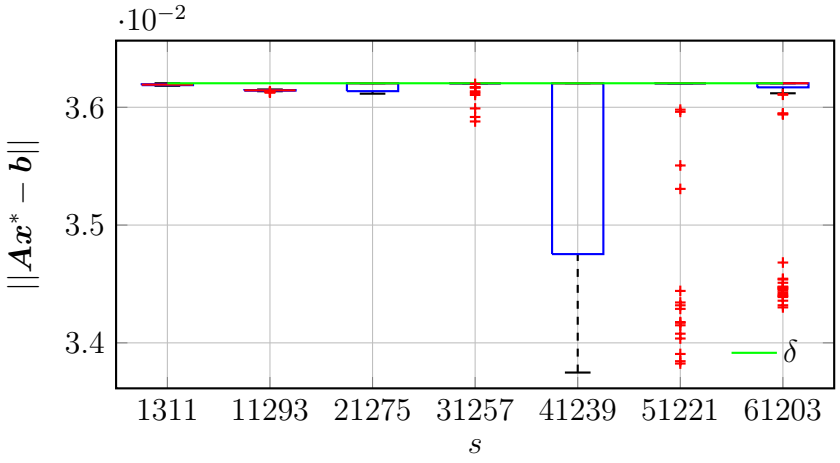
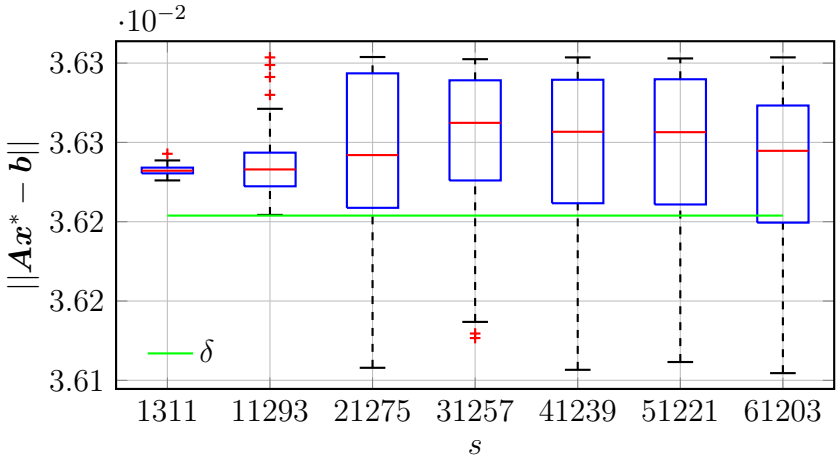


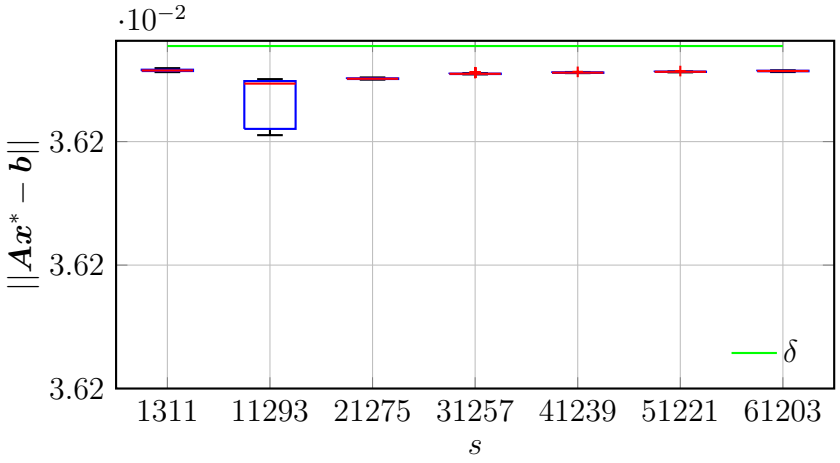
Figure 4.18: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noisy signals of 40 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.



(a)



(b)



(c)

Figure 4.19: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noisy signals of 80 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.

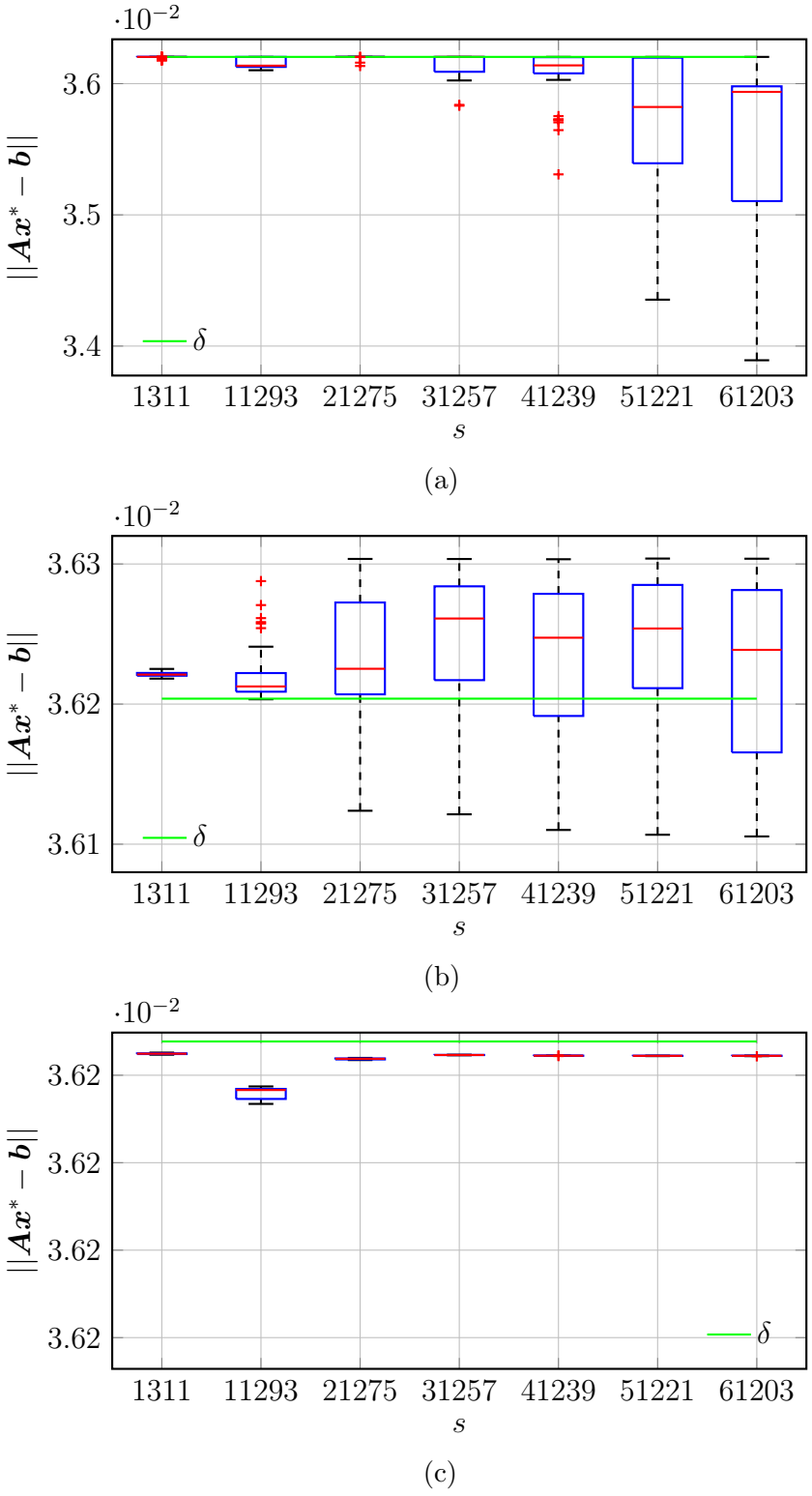


Figure 4.20: Box plot of $||\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}||$ for noisy signals of 100 dB: (a) FIPPP, (b) SPGL1, and (c) NESTA methods.

Chapter 5

Conclusions and Future Work

5.1 Introduction

Several methods for compressive sensing (CS) that can be used to solve a wide range of problems have been proposed. Compressible signals are recovered from a very limited number of measurements by minimizing nonconvex sparsity-promoting functions (SPFs) that closely resemble the ℓ_0-norm function. The proposed methods fall into two categories, namely, sequential convex formulation (SCF) and proximal-point (PP) based methods. Proposed SCF methods include the quadratic approximation (QA)-smoothly-clipped absolute deviation (SCAD), the piecewise-linear approximation (PLA)-SCAD methods of Chapter 2, and the $\mathcal{P}$-class family of methods of Chapter 3. Proposed PP methods include the iterative proximal-point projection (IPPP) and the *fast* iterative proximal-point projection (FIPPP) methods of Chapter 4. The proposed and corresponding competing methods were evaluated in terms of their capability in recovering sparse signals from Gaussian and orthonormal ensembles in a wide range of test problems. Simulation results demonstrate that the proposed methods lead to shorter more compact signal representations than those obtained with competing methods while requiring a comparable amount of computation.

5.2 Conclusions

In the QA-SCAD method, a solution of the recovery problem is approached by employing the QA of the SCAD function. Convex subproblems are solved by using a second-order solver (SOS) where the Newton step is computed efficiently. A target

value of the regularization term of the recovery problem is approached efficiently by using a continuation procedure. In the PLA-SCAD method, a solution to the recovery problem is approached by employing a PLA of the SCAD function. Convex subproblems are reformulated as second-order cone programming (SOCP) problems and are solved efficiently by using state-of-the-art SOSs such as the self-dual-minimization (SeDuMi) method. Simulation results demonstrate that the proposed methods achieve superior reconstruction performance (RP) metrics in terms of increased probability of perfect recovery (PPR) and reduced minimum required fraction (MRF) for perfect recovery when compared with competing regularized least-squares (RLS) and basis pursuit (BP) methods. The computational cost (CC) metric of the QA-SCAD method was found to be comparable to those of the competing methods, namely, the gradient projection for sparse reconstruction (GPSR) method of Figueiredo, Nowak, and Wright and the ℓ_1-LS method of Kim, Koh, Lustig, Boyd, and Gorinevsky. On the other hand, the CC metric of the PLA-SCAD method is increased relative to those of the competing methods, namely, the ℓ_1-Magic method of Candès and Romberg and the spectral projected-gradient ℓ_1-norm (SPGL1) method of Berg and Friedlander.

In the $\mathcal{P}$-class family of methods, the solution of the recovery problem is approached by employing a PLA of a $\mathcal{P}$-class function. Results obtained pertaining to the optimality conditions of the $\mathcal{P}$-class problems employed show that their local minimizers are sparse points. Convex subproblems are formulated as weighted ℓ_1-norm minimization problems while an efficient first-order solver (FOS) based on the NESTA method is employed. The proposed solver is capable of using the matrices involved in matrix-vector operations only. The sequence of solution points was shown to be a monotonically decreasing sequence of values of the objective function and converges to a sparse local minimizer of a $\mathcal{P}$-class problem. Simulation results demonstrate that the proposed methods are robust, lead to fast convergence, and achieve superior RP metrics in terms of increased PPR, reduced MRF for perfect recovery, and reduced average ℓ_∞ reconstruction error when compared with the ℓ_1-Magic method of Candès and Romberg, the iteratively reweighted least squares (IRWLS) method of Chartrand and Yin., the SPGL1 method of Berg and Friedlander, and the difference-of-two-convex-functions (DC)-family of methods of Gasso, Rakotomamonjy, and Canu. CC metric, in terms of the average CPU time, of the new methods was found to be comparable to that of the SPGL1 method and reduced as compared to those of the ℓ_1-Magic, IRWLS, and DC-family methods.

In the IPPP and FIPPP methods, the recovery process is carried out by minimiz-

ing the sum of the ϵ-ℓ_p^p norm function and the indicator function of a closed ball under affine mapping. The objective function obtained in this way exhibits rich properties such as a convex and differentiable Moreau envelope (ME) and a cohypomonotone subgradient mapping. When the iteration sequence is computed approximately, the method is applied by iteratively performing two fundamental operations, namely, computation of the PP of the ϵ-ℓ_p^p norm function and projection of the PP onto the closed ball under affine mapping. The first operation boils down to finding the largest real root of a trinomial equation which can be performed analytically or numerically as the limit of an infinite series of nested radicals. The second operation boils down to finding a point in the intersection of convex sets and can be performed efficiently by computing a sequence of closed-form projectors while using the matrices involved in matrix-vector operations only. The error sequence associated with the approximate computation is shown to be monotonically decreasing and summable. Consequently, the sequence of points associated with the iterative computation converges to a minimizer. Accelerated convergence is achieved by using a two-step method with optimal convergence rate. Simulation results demonstrate that the proposed methods achieve superior RP metrics such as increased PPR, reduced MRF for perfect recovery, and reduced average ℓ_∞ reconstruction error relative to the SPGL1 method of Berg and Friedlander and the NESTA method of Becker, Bobin, and Candès. In addition, superior measurement consistency (MC) metric are achieved when using the proposed method relative to the competing methods. CC metrics such as the average CPU time and number of matrix-vector operations were found to be comparable to those of the competing methods.

Some recommendations pertaining to the proposed methods will now be highlighted. In this discussion, small-, medium-, large-, and very-large-scale loosely apply to test problems with the number of samples of the sparse signals recovered, n, in the ranges $n < 2^{12}$, $2^{12} < n < 2^{16}$, $2^{16} < n < 2^{18}$, and $n > 2^{18}$, respectively. The QA-SCAD and PLA-SCAD methods are recommended for applications where small signals need to be recovered very accurately. As these methods are based on SOSs, very accurate solutions can be obtained but their use becomes problematic in medium-, large-, and very-large-scale problems as they are required to solve large systems of linear equations in computing the Newton step. In addition, the solution of these systems entails the storage and manipulation of large matrices. The $\mathcal{P}$-class family of methods is recommended for applications where large signals need to be recovered while employing a wide range of SPFs. These methods are based on an FOS that is

capable of using the matrices involved in matrix-vector operations only. Therefore, there is no need for the storage of matrices and matrix-vector operations can be carried out with fast algorithms such as the fast Fourier transform (FFT) when using orthogonal ensembles. The $\mathcal{P}$-class family of methods is inefficient for the solution of very-large-scale problems as these methods entail the solution of several subproblems of the same scale in sequence. The IPPP and FIPPP methods are recommended in applications where very-large signals need to be recovered accurately and efficiently.

5.3 Future Work

Speeding up the convergence of SCF methods is an important future research direction. Results pertaining to these methods have shown that when perfect signal recovery is achieved, four subproblems are solved on average for the sequence of solution points involved to converge. This is typically prohibitive for solving very-large-scale problems. Hence, the applicability of SCF methods, such as the $\mathcal{P}$-class family of methods of Chapter 3, can be extended to problems of larger scale if the average number of subproblems solved can be reduced.

The application of PP based methods for the solution of more general recovery problems is another interesting future research direction. For example, the ϵ-ℓ_p^p norm function approaches the ℓ_0 norm when $p \to 0$ and $\epsilon \to 0$. Hence, improved recovery performance is achieved if the applicability of the IPPP and FIPPP methods of Chapter 4 can be extended to the case where $p < \frac{1}{2}$. More general recovery problems, such as the $\mathcal{P}$-class problems of Chapter 3, could also be used to achieve improved recovery performance.

If compressible signals are recovered by minimizing convex SPFs such as the ℓ_1-norm function, any local minimizer is also the sparsest signal representation when the number of nonzero-valued samples of the recovered signal and certain matrix conditions are met. Unfortunately, to our knowledge, similar conditions are not known to exist when nonconvex SPFs are employed. Nevertheless, numerical evidence appears to suggest that the local minimizers of nonconvex SPFs addressed in this dissertation have similar properties as their convex counterparts. This unusual property suggests that the problems examined belong to the so-called *class of hidden-convex optimization problems* identified by Wu, Li, Zhang, and Yang. Motivated by the experimental results presented in this dissertation, investigating this hypothesis is another interesting future research direction.

Bibliography

[1] The MOSEK Optimization Tools Version 2.5. User's manual and reference, 2002.

[2] R. Aharoni and Y. Censor. Block-iterative projection methods for parallel computation of solutions to convex feasibility problems. *Linear Algebra and its Applications*, 120(0):165–175, 1989.

[3] E. L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. SIAM, 2003.

[4] A. Antoniou and W.-S. Lu. *Practical Optimization, Algorithms and Engineering Applications*. Springer Science+Business Media, 2007.

[5] M.S. Asif, W. Mantzel, and J. Romberg. Random channel coding and blind deconvolution. In *Allerton Conference on Communication, Control, and Computing*, pages 1021–1025, October 2009.

[6] H. Bauschke, P. L. Combettes, and S. Kruk. Extrapolation algorithm for affine-convex feasibility problems. *Numerical Algorithms*, 41:239–274, 2006.

[7] M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, 1979.

[8] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[9] S. Becker, J. Bobin, and E. J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[10] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[11] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

[12] D. P. Bertsekas. *Convex Optimization Theory.* Athena Scientific, 2009.

[13] D. P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning.* MIT Press, 2011.

[14] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples, Second Edition.* Springer Science+Business Media, 2006.

[15] P. Boufounos and R. Baraniuk. Quantization of sparse representations. In *Data Compression Conference*, page 378, 2007.

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

[17] K. Bredies and D. Lorenz. Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications*, 14:813–837, 2008.

[18] E. J. Candès and J. Romberg. ℓ_1*-Magic: Recovery of Sparse Signals via Convex Programming*, 2005.

[19] E. J. Candès and J. Romberg. Encoding the ℓ_p ball from limited measurements. In *Data Compression Conference*, pages 33–42, 2006.

[20] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, feb. 2006.

[21] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223, 2006.

[22] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory*, 52(12):5406–5425, 2006.

[23] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l1 minimization. *J. Fourier Anal. Appl.*, 14(5):877–905, 2008.

[24] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14(10):707 –710, October 2007.

[25] R. Chartrand. Fast algorithms for nonconvex compressive sensing: Mri reconstruction from very few data. In *IEEE International Symposium on Biomedical Imaging*, pages 262–265, 2009.

[26] R. Chartrand and V. Staneva. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems*, 24(3):1–14, 2008.

[27] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3869 –3872, 2008.

[28] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 2001.

[29] P. L. Combettes. Convex set theoretic image recovery with inexact projection algorithms. In *2001 International Conference on Image Processing*, volume 1, pages 257 –260, 2001.

[30] P. L. Combettes and T. Pennanen. Proximal methods for cohypomonotone operators. *SIAM Journal on Control and Optimization*, 43(2):731–742, 2004.

[31] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Model. Simul.*, 4(4):1168–1200, 2005.

[32] P.L. Combettes. Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections. *IEEE Transactions on Image Processing*, 6(4):493–506, 1997.

[33] P.L. Combettes. Hilbertian convex feasibility problem: Convergence of projection methods. *Applied Mathematics and Optimization*, 35(3):311–330, 1997.

[34] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. SIAM, 2000.

[35] A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

[36] J. Dattorro. *Convex Optimization & Euclidean Distance Geometry.* Meboo Publishing USA, 2011.

[37] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Gunturk. Iteratively reweighted least squares minimization for sparse recovery. *Comm. Pure Appl. Math.*, 63(1):1–38, 2010.

[38] A.G. Dimakis and P.O. Vontobel. Lp decoding meets lp decoding: A connection between channel coding and compressed sensing. In *Allerton Conference on Communication, Control, and Computing*, pages 8–15, October 2009.

[39] D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

[40] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32:407–499, 2004.

[41] K. Eriksson, D. Estep, and C. Johnson. Applied Mathematics: Body and Soul, Vol. 1. 2004.

[42] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statistical Assoc.*, 96(456):1348–1360, 2001.

[43] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Topics Signal Process.*, 1(4):586–597, 2007.

[44] R. Fletcher. *Practical methods of optimization; (Vol. 2).* Wiley-Interscience, New York, NY, USA, 1981.

[45] S. Foucart and M. T. Lai. Sparsest solutions of underdetermined linear systems via ℓ_q- minimization for $0 < q \leq 1$. *Appl. Comput. Harmon. Anal.*, 26(3):395–407, 2009.

[46] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *IEEE Trans. Signal Process.*, 57(12):4686 –4698, December 2009.

[47] D. Ge, X. Jiang, and Y. Ye. A note on the complexity of l_p minimization. *Mathematical Programming*, 129:285–299, 2011.

[48] A. M. Geoffrion. Objective function approximations in mathematical programming. *Mathematical Programming*, 13:23–37, 1977.

[49] J. R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in matlab: design and implementation. *SIAM J. Matrix Anal. Appl.*, 13(1):333–356, January 1992.

[50] D. Goldfarb and M. J. Todd. Optimization. chapter Linear programming, pages 73–170. Elsevier North-Holland, Inc., 1989.

[51] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *IEEE Trans. Signal Process.*, 45(3):600–616, 1997.

[52] L.G. Gubin, B.T. Polyak, and E.V. Raik. The method of projections for finding the common point of convex sets. {*USSR*} *Computational Mathematics and Mathematical Physics*, 7(6):1 – 24, 1967.

[53] C. Güntürk, M. Lammers, A. Powell, R. Saab, and Ö. Yilmaz. Sobolev duals for random frames and sigma-delta quantization of compressed sensing measurements. *Foundations of Computational Mathematics*, 13(1):1–36, 2013.

[54] W. W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):pp. 221–239, 1989.

[55] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 4th edition, 1960.

[56] W. Hare and C. Sagastizàbal. Computing proximal points of nonconvex functions. *Math. Program.*, 116(1):221–258, jun 2008.

[57] A. Herschfeld. On infinite radicals. *Amer. Math. Monthly*, 42:419–429, 1935.

[58] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[59] R. Horst. On the global minimization of concave functions. *OR Spectrum*, 6:195–205, 1984.

[60] R. Horst and N. V. Thoai. Dc programming: Overview. *Journal of Optimization Theory and Applications*, 103:1–43, 1999.

[61] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley & Sons, 2 edition, 2009.

[62] D.R. Hunter and K. Lange. A tutorial on mm algorithms. *The American Statistician*, 58:30–37, Feb 2004.

[63] A. N. Iusem, T. Pennanen., and B. F. Svaiter. Inexact variants of the proximal point algorithm without monotonicity. *SIAM Journal on Optimization*, 13(4):1080–1097, 2003.

[64] M.W. Jacobson and J.A. Fessler. An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Trans. Image Process.*, 16(10):2411 –2422, 2007.

[65] L. Jacques, J. Laska, P. Boufounos, and R. G. Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Trans. Inf. Theory*, 59(4):2082–2102, 2013.

[66] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[67] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Number 16 in Frontiers in Applied Mathematics. SIAM, 1995.

[68] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[69] F. Krahmer, R. Saab, and Ö. Yilmaz. Sigma-delta quantization of sub-gaussian frame expansions and its application to compressed sensing. *Information and Inference*, 3(1):40–58, 2014.

[70] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *J. Comput. Graph. Statist.*, 9(1):1–20, 2000.

[71] Kenneth Lange. *Numerical Analysis for Statisticians, Second Edition*. Springer Science+Business Media, 2010.

[72] C. L. Lawson. *Contributions to the Theory of Linear Least Maximum. Approximations*. PhD thesis, University of California, Los. Angeles, 1961.

[73] David G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., 1969.

[74] D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[75] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, December 2007.

[76] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3 edition, 2008.

[77] R. McGill, J. W. Tukey, and W. A. Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.

[78] J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la S. M. F.*, 93:273–299, 1965.

[79] R. J. Muirhead. *Aspects of Multivariate Statistical Theory*. John Wiley & Sons, 1982.

[80] B. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[81] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[82] Y. Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., 2004.

[83] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[84] A. Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Review*, 40(3):636–666, 1998.

[85] J. Nocedal and S. J. Wright. *Numerical Optimization, Second Edition*. Springer Science+Business Media, 2006.

[86] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.

[87] T. Pennanen. Local convergence of the proximal point algorithm and multiplier methods without monotonicity. *Mathematics of Operations Research*, 27(1):pp. 170–191, 2002.

[88] Y. Plan and R. Vershynin. One-bit compressed sensing by linear programming. *Commun. Pure Appl. Math.*, 66(8):1275–1297, 2013.

[89] R. A. Poliquin and R. T. Rockafellar. Prox-regular functions in variational analysis. *Trans. Amer. Math. Soc.*, 348(5):1805–1838, May 1996.

[90] B. T. Polyak. *Introduction to Optimization*. Optimization Software, Publications Division, 1987.

[91] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[92] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. K.-Delgado. Subset selection in noise based on diversity measure minimization. *IEEE Trans. Signal Process.*, 51(3):760–770, 2003.

[93] B. D. Rao and K. Kreutz-Delgado. An affine scaling methodology for best basis selection. *IEEE Trans. Signal Process.*, 47(1):187–200, 1999.

[94] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[95] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

[96] R. T. Rockafellar and R.-J. B. Wets. *Variational Analysis*. Springer, 1997.

[97] R. Saab and Ö. Yilmaz. Sparse recovery by non-convex optimization instance optimality. *Appl. Comput. Harmon. Anal.*, 29(1):30–48, 2010.

[98] F. Santosa and W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[99] D. Sarafyan. Nested series, computation of square roots and solution of third degree equations. *Mathematics Magazine*, 27(1):pp. 19–36, 1953.

[100] J. Stewart. *Calculus*. Thomson Brooks/Cole, 2003.

[101] Gilbert Strang. *Linear Algebra and Its Applications*. Thomson Brooks/Cole, 4 edition, 2006.

[102] J. F. Sturm. Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones, 1998.

[103] P. G. Szabó. On the roots of the trinomial equation. *Central European Journal of Operations Research*, 18(1):97–104, 2010.

[104] F. C. A. Teixeira, S. W. Bergen, and A. Antoniou. Robust signal recovery approach for compressive sensing using unconstrained optimization. In *Proc. IEEE Int. Symp. Circuits and Systems*, pages 3521 –3524, 2010.

[105] F. C. A. Teixeira, S. W. A. Bergen, and A. Antoniou. Signal recovery method for compressive sensing using relaxation and second-order cone programming. In *Proc. IEEE Int. Symp. Circuits and Systems*, pages 2125–2128, 2011.

[106] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

[107] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[108] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. Beyond nyquist: Efficient sampling of sparse, bandlimited signals. *IEEE Trans. Inf. Theory*, 56(1):520–544, 2010.

[109] J. Trzasko and A. Manduca. Relaxed conditions for sparse signal recovery with general concave priors. *IEEE Trans. Signal Process.*, 57(11):4347 –4354, November 2009.

[110] J. V. Uspensky. *Theory of Equations*. McGraw-Hill, 1948.

[111] K. Xiangming, P. Petre, R. Matic, A.C. Gilbert, and M.J. Strauss. An analog-to-information converter for wideband signals using a time encoding machine. In *IEEE DSP/SPE*, pages 414 –419, jan. 2011.

[112] E. Yildirim and S. Wright. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12(3):782–810, 2002.

[113] J. Yoo, S Becker, M. Monge, M. Loh, E. J. Candès, and A. Emami-Neyestanak. Design and implementation of a fully integrated compressed-sensing signal acquisition system. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2012.

[114] D. C. Youla. Mathematical theory of image restoration by the method of convex projections. In H. Stark, editor, *Image Recovery: Theory and Application*. Academic Press, 1987.

[115] W. I. Zangwill. *Nonlinear Programming: A Unified Approach*. Prentice-Hall, 1969.

[116] C. Zălinescu. *Convex Analysis in General Vector Spaces*. World Scientific, 2002.