SIAM J. OPTIM.
Vol. 30, No. 3, pp. 2410–2440

AN ASYMPTOTICALLY SUPERLINEARLY CONVERGENT SEMISMOOTH NEWTON AUGMENTED LAGRANGIAN METHOD FOR LINEAR PROGRAMMING*

XUDONG LI†, DEFENG SUN‡, AND KIM-CHUAN TOH§

Abstract. Powerful interior-point methods (IPM) based commercial solvers, such as Gurobi and Mosek, have been hugely successful in solving large-scale linear programming (LP) problems. The high efficiency of these solvers depends critically on the sparsity of the problem data and advanced matrix factorization techniques. For a large scale LP problem with data matrix A that is dense (possibly structured) or whose corresponding normal matrix AA^T has a dense Cholesky factor (even with reordering), these solvers may require excessive computational cost and/or extremely heavy memory usage in each interior-point iteration. Unfortunately, the natural remedy, i.e., the use of iterative methods based IPM solvers, although it can avoid the explicit computation of the coefficient matrix and its factorization, is often not practically viable due to the inherent extreme ill-conditioning of the large scale normal equation arising in each interior-point iteration. While recent progress has been made to alleviate the ill-conditioning issue via sophisticated preconditioning techniques, the difficulty remains a challenging one. To provide a better alternative choice for solving large scale LPs with dense data or requiring expensive factorization of its normal equation, we propose a semismooth Newton based inexact proximal augmented Lagrangian (SNIPAL) method. Different from classical IPMs, in each iteration of SNIPAL, iterative methods can efficiently be used to solve simpler yet better conditioned semismooth Newton linear systems. Moreover, SNIPAL not only enjoys a fast asymptotic superlinear convergence but is also proven to enjoy a finite termination property. Numerical comparisons with Gurobi have demonstrated encouraging potential of SNIPAL for handling large-scale LP problems where the constraint matrix A has a dense representation or AA^T has a dense factorization even with an appropriate reordering. For a few large LP instances arising from correlation clustering, our algorithm can be up to 20–100 times faster than the barrier method implemented in Gurobi for solving the problems to the accuracy of 10^{-8} in the relative KKT residual. However, when tested on some large sparse LP problems available in the public domain, our algorithm is not yet practically competitive against the barrier method in Gurobi, especially when the latter can compute the Schur complement matrix and its sparse Cholesky factorization in each iteration cheaply.

Key words. linear programming, semismooth Newton method, augmented Lagrangian method

AMS subject classifications. 90C05, 90C06, 90C25, 65F10

DOI. 10.1137/19M1251795

1. Introduction.

It is well known that primal-dual interior-point methods (IPMs) as implemented in highly optimized commercial solvers, such as Gurobi and Mosek, are powerful methods for solving large scale linear programming (LP) problems with conducive sparsity. However, the large scale normal (also called Schur complement)

*Received by the editors March 22, 2019; accepted for publication (in revised form) June 11, 2020; published electronically September 8, 2020.
https://doi.org/10.1137/19M1251795
Funding: The first author's research is supported by the National Natural Science Foundation of China (11901107), the Young Elite Scientists Sponsorship Program by CAST (2019QNRC001), and the Shanghai Sailing Program (19YF1402600). The second author's research is partially supported by The Hong Kong Polytechnic University under the 2017 Postdoctoral Fellowships Scheme. The third author's research is partially supported by the Academic Research Fund of the Ministry of Singapore under grant R146-000-257-112.

†School of Data Science, Fudan University, Shanghai, China; Shanghai Center for Mathematical Sciences, Fudan University, Shanghai, China (lixudong@fudan.edu.cn).
‡Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (defeng.sun@polyu.edu.hk).
§Department of Mathematics, and Institute of Operations Research and Analytics, National University of Singapore, Singapore (mattohkc@nus.edu.sg).

equation arising in each interior-point iteration is generally highly ill-conditioned when the barrier parameter is small, and typically it is necessary to employ a direct method, such as the sparse Cholesky factorization, to solve the equation stably and accurately. Various attempts, for example, in [3, 10, 15, 24, 34], have been made in using an iterative solver, such as the preconditioned conjugate-gradient (PCG) method, to solve the normal equation when it is too expensive to compute the coefficient matrix or the sparse Cholesky factorization because of excessive computing time or memory usage due to fill-ins. For more details on the numerical performance of iterative methods based IPMs for solving large scale LP, we refer readers to [15] and the references therein. However, the extreme ill-conditioning of the normal equation (and also of the augmented equation) makes it extremely costly for an iterative method to solve the equation either because it takes an excessive number of steps to converge or because constructing an effective preconditioner is prohibitively expensive. For a long time since their inceptions, iterative methods based IPMs have not been proven convincingly to be more efficient in general than the highly powerful solvers, such as Gurobi and Mosek, on various large scale LP test instances. Fortunately, recently promising progress has been made in the work of Schork and Gondzio [45], where the authors proposed effective basis matrix preconditioners for iterative methods based IPMs, which have been demonstrated to be competitive against the powerful commercial solver Gurobi on some large scale LPs in MIPLIB [33]. However, we should note that as the construction of the basis matrix preconditioners in [45] requires the explicit storage of a subset of columns of the constraint matrix A, the approach may not be applicable to the case when A is not explicitly given but defined via a linear map. In contrast, the algorithm designed in this paper is still applicable under the latter scenario. While this paper was in the final review, the preprint [4] appeared, where the authors proposed a potentially cheaper preconditioning approach, compared to those in [45], for regularized interior point methods for linear and convex quadratic programming (QP). But the numerical performance of the new approach in [4] is not compared against Gurobi.

For later discussion, here we give an example where A is defined by a linear map: $A \in \mathbb{R}^{n^2} \to \mathbb{R}^{p^2}$ such that $Ax = \mathrm{vec}(B\,\mathrm{mat}(x)\,D^T)$, where $B, D \in \mathbb{R}^{p\times n}$ are given matrices, $\mathrm{mat}(x)$ denotes the operation of converting a vector $x \in \mathbb{R}^{n^2}$ into an $n \times n$ matrix, and $\mathrm{vec}(X)$ denotes the operation of converting a matrix $X \in \mathbb{R}^{p\times p}$ into a p^2-dimensional vector. It is easy to see that the matrix representation of A is the Kronecker product $D \otimes B$, and it could be extremely costly to store $D \otimes B$ explicitly when B, D are large dimensional dense matrices.

The goal of this paper is to design a semismooth Newton inexact proximal augmented Lagrangian (Snipal) method for solving large scale LP problems, which has the following key properties: (a) the Snipal method can achieve fast local linear convergence; (b) the semismooth Newton equation arising in each iteration can fully exploit the solution sparsity in addition to data sparsity; (c) the semismooth Newton equation is typically much better conditioned than its counterparts in IPMs, even when the iterates approach optimality. The latter two properties thus make it cost effective for one to use an iterative method, such as the PCG method, to solve the aforementioned linear system when it is large. It is these three key properties that give the competitive advantage of our Snipal method over the highly developed IPMs for solving certain classes of large scale LP problems which we will describe shortly.

Consider the following primal and dual LP problems:

$$(\mathrm{P})\ \min\Big\{c^Tx+\delta_K(x)\mid Ax=b,\ x\in\mathbb{R}^n\Big\},$$

$$(\mathrm{D})\ \max\Big\{-\delta_K^*(A^*y-c)+b^Ty\mid y\in\mathbb{R}^m\Big\},$$

where $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$, $c\in\mathbb{R}^n$ are given data. The set $K=\{x\in\mathbb{R}^n\mid l\le x\le u\}$ is a simple polyhedral set, where l,u are given vectors. We allow the components of l and u to be $-\infty$ and ∞, respectively. In particular, K can model the nonnegative orthant $\mathbb{R}^n_+$. In the above, $\delta_K(\cdot)$ denotes the indicator function over the set K such that $\delta_K(x)=0$ if $x\in K$ and $\delta_K(x)=\infty$ otherwise. The Fenchel conjugate of δ_K is denoted by δ_K^*. We note that while we focus on the indicator function $\delta_K(\cdot)$ in (P), the algorithm and theoretical results we have developed in this paper are also applicable when δ_K is replaced by a closed convex polyhedral function $p:\mathbb{R}^n\to(-\infty,\infty]$. We make the following assumption on the problems (P) and (D).

Assumption 1. *The solution set of (P) and (D) is nonempty and A has full row rank (hence $m\le n$).*

Our Snipal method is designed for the dual LP but the primal variable is also generated in each iteration. In order for the fast local convergence property to kick-in early, we warm-start the Snipal method by an alternating direction method of multipliers (ADMM), which is also applied to the dual LP. We should mention that our goal is not to use Snipal as a general purpose solver for LP but to complement the excellent general solvers (Gurobi and Mosek) when the latter are too expensive or have difficulties in solving very large scale problems due to memory limitation. In particular, we are interested in solving large scale LP problems having one of the following characteristics.

1. The number of variables n in (P) is significantly larger than the number of linear constraints m. We note that such a property is not restrictive since for a primal problem with a huge number of inequality constraints $Ax\le b$ and $m\gg n$, we can treat the dual problem (D) as the primal LP, and the required property is satisfied.
2. The constraint matrix A is large and dense but it has an economical representation such as being the Kronecker product of two matrices, or A is sparse but AA^T has a dense factorization even with an appropriate reordering. For such an LP problem, it may not be possible to solve it by using the standard interior-point methods implemented in Gurobi or Mosek since A cannot be stored explicitly. Instead, one would need to use a Krylov subspace iterative method to solve the underlying large and dense linear system of equations arising in each iteration of an IPM or Snipal.

In [50], Wright proposed an algorithm for solving the primal problem (P) for the special case where $K=\mathbb{R}^n_+$. The proposed method is in fact the proximal method of multipliers applied to (P) while keeping the nonnegative constraint in the QP subproblem. More specifically, suppose that the iterate at the kth iteration is (x^k,y^k) and the penalty parameter is $\gamma_k=\sigma_k^{-1}$. Then the QP subproblem is given by $\min\{\frac{1}{2}\langle(\sigma_kA^*A+\sigma_k^{-1}I_n)x,\,x\rangle+\langle x,\,c-A^*y^k-\sigma_k^{-1}x^k-\sigma_kA^*b\rangle\mid x\ge0\}$. In [50], a successive over-relaxation (SOR) method is used to solve the QP subproblem. But it is unclear how this subproblem can be solved efficiently when n is large. In contrast, in this paper, we propose a Snipal method that is applied to the dual problem (D) and the associated subproblems are solved efficiently by

a semismooth Newton method having at least local superlinear convegence or even quadratic convergence.

In the pioneering work of De Leone and Mangasarian [30], an augmented Lagrangian method is applied to an equivalent reformulation of (D), and the QP subproblem of the form $\min\{-b^Ty + \frac{\sigma}{2}\|A^*y + z - c + \sigma^{-1}x^k\|^2 \mid y \in \mathbb{R}^m, z \geq 0\}$ in each iteration is solved by a projected SOR method. Interestingly, in a later paper [32], based on the results obtained in [31], Mangasarian designed a generalized Newton method to first solve a penalty problem of the form $\min\{-\epsilon b^Ty + \frac{1}{2}\|\Pi_{\mathbb{R}^n_+}(A^*y - c)\|^2\}$ and then use its solution to indirectly solve (P) for $K = \mathbb{R}^n_+$, under the condition that the positive parameter ϵ must be below a certain unknown threshold and a strong uniqueness condition holds. Soon after, [19] observed that the restriction on the parameter in [32] can be avoided by modifying the procedure in [32] via the augmented Lagrangian method but the corresponding subproblem in each iteration must be solved *exactly*. As the generalized Newton system is likely to be singular, in both [32] and [19], the system is modified by adding a scalar multiple of the identity matrix to the generalized Hessian. Such a perturbation, however, would destroy the fast local convergence property of the generalized Newton method. We also note that to obtain the minimum norm solution of the primal problem (P), [26] proposed a generalized Newton method for solving $\min\{\frac{1}{2}\|\Pi_{\mathbb{R}^n_+}(A^*y - rc)\|^2 - \langle b, y\rangle\}$ with the positive parameter r being sufficiently large. Although [26] contains no computational results, the authors obtained the global convergence and finite termination properties of the proposed method under the assumption that the Newton linear systems involved are solved exactly and a certain regularity condition on the nonsingularity of generalized Jacobians holds. More recently, [52] designed an ALM for the primal problem (P) for which a bound-constrained convex QP subproblem must be solved in each iteration. In the paper, this subproblem is solved by a randomized coordinate descent (RCD) method with an active set implementation. There are several drawbacks to this approach. First, solving the QP subproblem can be time consuming since the convergence of the RCD is generally quite slow. Second, the RCD approach is less effective in fully exploiting any specific structure of the matrix A (for example, when A is defined by the Kronecker product of two given matrices) to speed up the computation of the QP subproblem. Finally, it also does not exploit the sparsity structure present in the Hessian of the underlying QP subproblem to speed up the computation.

Here, we employ an inexact proximal augmented Lagrangian (PAL) method to (D) to simultaneously solve (P) and (D). Our entire algorithmic design is dictated by the focus on computational efficiency and generality. From this perspective, now we elaborate on the key differences between our paper and [19]. First, without any reformulation, our algorithm is directly applicable to problems with a more general set K instead of just $\mathbb{R}^n_+$ as in [32] and [19]. Second, we use the inexact PAL framework, which ensures that in each iteration, an unconstrained minimization subproblem involving the variable y is strongly convex and hence the semismooth Newton method we employ to solve this subproblem can attain local quadratic convergence. Third, the flexibility of allowing the PAL subproblems to be solved inexactly can lead to substantial computational savings, especially during the initial phase of the algorithm. Fourth, for computational efficiency, we warm-start our inexact PAL method by using a first-order method. Finally, as solving the semismooth Newton linear systems is the most critical component of the entire algorithm, we have devoted a substantial part of the paper to proposing novel numerical strategies to solve the linear systems efficiently.

Numerical comparisons of our SNIPAL with the barrier method in Gurobi have demonstrated the encouraging potential of our method for handling large-scale LP problems where the constraint matrix A has a dense representation or AA^T has a dense factorization even with an appropriate reordering. For a few large LP instances arising from correlation clustering, our algorithm can be up to 20–100 times faster than the barrier method implemented in Gurobi for solving the problems to the accuracy of 10^{-8} in the relative KKT residual. However, when tested on some large sparse LP problems available in the MIPLIB2010 [33], our algorithm is not yet practically competitive against the barrier method in Gurobi, especially when the latter can compute the Schur complement matrix and its sparse Cholesky factorization in each iteration cheaply.

The remaining part of the paper is organized as follows. In the next section, we introduce a preconditioned proximal point algorithm (PPA) and establish its global and local (asymptotic) superlinear convergence. In section 3, we develop a SNIPAL for solving the dual LP (D) and derive its connection to the preconditioned PPA. Section 4 is devoted to developing numerical techniques for solving the linear system of equations in the semismooth Newton method employed to solve the subproblem in each proximal augmented Lagrangian iteration. We describe how to employ an ADMM to warm-start the proximal augmented Lagrangian method in section 5. In section 6, we evaluate the numerical performance of our algorithm (called SNIPAL) against the barrier method in Gurobi on various classes of large scale LPs, including some large sparse LPs available in the public domain. We conclude the paper in the final section.

Notation. We use $\mathcal{X}$ and $\mathcal{Y}$ to denote finite dimensional real Euclidean spaces each endowed with an inner product $\langle\cdot,\,\cdot\rangle$ and its induced norm $\|\cdot\|$. For any self-adjoint positive semidefinite linear operator $\mathcal{M}:\mathcal{X}\to\mathcal{X}$, we define $\langle x,\,x'\rangle_{\mathcal{M}}:=\langle x,\,\mathcal{M}x'\rangle$ and $\|x\|_{\mathcal{M}}:=\sqrt{\langle x,\,\mathcal{M}x\rangle}$ for all $x,x'\in\mathcal{X}$. The largest eigenvalue of $\mathcal{M}$ is denoted by $\lambda_{\max}(\mathcal{M})$. A similar notation is used when $\mathcal{M}$ is replaced by a matrix M. Let D be a given subset of $\mathcal{X}$. We write the weighted distance of $x\in\mathcal{X}$ to D by $\mathrm{dist}_{\mathcal{M}}(x,D):=\inf_{x'\in D}\|x-x'\|_{\mathcal{M}}$. If $\mathcal{M}$ is the identity operator, we just omit it from the notation so that $\mathrm{dist}(\cdot,D)$ is the Euclidean distance function. If D is closed, the Euclidean projector over D is defined by $\Pi_D(x):=\mathrm{argmin}\{\|x-d\|\mid d\in D\}$. Let $F:\mathcal{X}\rightrightarrows\mathcal{Y}$ be a multivalued mapping. We define the graph of F to be the set $\mathrm{gph}F:=\{(x,y)\in\mathcal{X}\times\mathcal{Y}\mid y\in F(x)\}$. The range of a multifunction is defined by $\mathrm{Range}(F):=\{y\mid\exists\,x\text{ with }y\in F(x)\}$.

2. A preconditioned proximal point algorithm. In this section, we present a preconditioned PPA and study its convergence properties. In particular, following the classical framework developed in [41, 42], we prove the global convergence of the preconditioned PPA. Under a mild error bound condition, global linear rate convergence is also derived. In fact, by choosing the parameter c_k in the algorithm to be sufficiently large, the linear rate can be as fast as we please. We further show in section 3.1 that our main algorithm, SNIPAL, is in fact an application of the preconditioned PPA. Hence, SNIPAL's convergence properties can be obtained as a direct application of the general theory developed here.

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite dimensional Hilbert spaces and $\mathcal{T}:\mathcal{X}\to\mathcal{X}$ be a maximal monotone operator. Throughout this section, we assume that $\Omega:=\mathcal{T}^{-1}(0)$ is nonempty. We further note from [43, Excerise 12.8] that Ω is a closed set. The preconditioned PPA generates for any start point $z^0\in\mathcal{X}$ a sequence $\{z^k\}\subseteq\mathcal{X}$ by the following

approximate rule:

$$z^{k+1} \approx \mathcal{P}_k(z^k), \quad \text{where} \quad \mathcal{P}_k = (\mathcal{M}_k + c_k\mathcal{T})^{-1}\mathcal{M}_k. \tag{2.1}$$

Here $\{c_k\}$ and $\{\mathcal{M}_k\}$ are some sequences of positive real numbers and self-adjoint positive definite linear operators over $\mathcal{X}$. If $\mathcal{M}_k \equiv \mathcal{I}$ for all $k \geq 0$, the updating scheme (2.1) recovers the classical PPA considered in [41]. Since $\mathcal{M}_k + c_k\mathcal{T}$ is a strongly monotone operator, we know from [43, Proposition 12.54] that $\mathcal{P}_k$ is single-valued and is globally Lipschitz continuous. Here, we further assume that $\{c_k\}$ bounded away from zero and

$$(1+\nu_k)\mathcal{M}_k \succeq \mathcal{M}_{k+1}, \quad \mathcal{M}_k \succeq \lambda_{\min}\mathcal{I} \quad \forall\, k \geq 0 \quad \text{and} \ \limsup_{k\to\infty} \lambda_{\max}(\mathcal{M}_k) = \lambda_\infty \tag{2.2}$$

with some nonnegative summarable sequence $\{\nu_k\}$ and constants $+\infty > \lambda_\infty \geq \lambda_{\min} > 0$. The same condition on $\mathcal{M}_k$ is also used in [36] and can be easily satisfied. For example, it holds obviously if we set $\lambda_\infty\mathcal{I} \succeq \mathcal{M}_k \succeq \lambda_{\min}\mathcal{I}$ and $\mathcal{M}_k \succeq \mathcal{M}_{k+1}$ for all $k \geq 0$. Note that if $\mathcal{T}$ is a linear operator, one may rewrite $\mathcal{P}_k$ as $\mathcal{P}_k = (\mathcal{I}+c_k\mathcal{M}_k^{-1}\mathcal{T})^{-1}$. We show in the next lemma that this expression in fact holds even for a general maximal monotone operator $\mathcal{T}$. Therefore, we can regard the self-adjoint positive definite linear operator $\mathcal{M}_k$ as a preconditioner for the maximal monotone operator $\mathcal{T}$. Based on this observation, we name the algorithm described in (2.1) as the preconditioned PPA.

LEMMA 2.1. *Given a constant $\alpha > 0$, a self-adjoint positive definite linear operator $\mathcal{M}$, and a maximal monotone operator $\mathcal{T}$ on $\mathcal{X}$, it holds that* $\mathrm{Range}(\mathcal{I}+\alpha\mathcal{M}^{-1}\mathcal{T}) = \mathcal{X}$ *and $(\mathcal{I} + \alpha\mathcal{M}^{-1}\mathcal{T})^{-1}$ is a single-valued mapping. In addition,*

$$(\mathcal{M} + \alpha\mathcal{T})^{-1}\mathcal{M} = (\mathcal{I} + \alpha\mathcal{M}^{-1}\mathcal{T})^{-1}.$$

Proof. By [2, Proposition 20.24], we know that $\mathcal{M}^{-1}\mathcal{T}$ is maximally monotone. Hence, $\mathrm{Range}(\mathcal{I} + \alpha\mathcal{M}^{-1}\mathcal{T}) = \mathcal{X}$ and $(\mathcal{I} + \alpha\mathcal{M}^{-1}\mathcal{T})^{-1}$ is a single-valued mapping from $\mathcal{X}$ to itself.

Now, for any given $z \in \mathcal{X}$, suppose that $z_1 = (\mathcal{I} + \alpha\mathcal{M}^{-1}\mathcal{T})^{-1}(z)$. Then, it holds that

$$\mathcal{M}z \in (\mathcal{M} + \alpha\mathcal{T})z_1.$$

Since $(\mathcal{M} + \alpha\mathcal{T})^{-1}$ is a single-valued operator [43, Proposition 12.54], we know that

$$z_1 = (\mathcal{M} + \alpha\mathcal{T})^{-1}\mathcal{M}z,$$

i.e., $(\mathcal{I} + \alpha\mathcal{M}^{-1}\mathcal{T})^{-1}z = (\mathcal{M} + \alpha\mathcal{T})^{-1}\mathcal{M}z$ for all $z \in \mathcal{X}$. Thus we have proved the desired equation. ∎

In the literature, the updating scheme (2.1) is closely related to the so-called variable metric PPAs; for examples, see [6, 8, 7, 9, 13, 36, 37]. Among these papers, [6, 13, 37] focus only on the case of optimization, i.e., the maximal monotone operator $\mathcal{T}$ is the subdifferential mapping of a convex function. In addition, they emphasize more on the combination of the PPA with the quasi Newton method. In [8] and the subsequent papers [7, 9], the authors deal with a general maximal monotone operator $\mathcal{T}$ and study the following scheme in the exact setting:

$$z^{k+1} = z^k + \mathcal{M}_k\big((\mathcal{I} + c_k\mathcal{T})^{-1} - \mathcal{I}\big)z^k. \tag{2.3}$$

The global convergence of the scheme (2.3) requires a rather restrictive assumption on

$\mathcal{M}_k$ [8, Hypothesis (H2)], although $\mathcal{M}_k$ is not required to be self-adjoint. In fact, the authors essentially assumed that the deviation of $\mathcal{M}_k$ from the identity operator should be small, and the verification of the assumption can be quite difficult. As far as we are aware of, [36] may be the most related work to ours. In [36], the authors consider a variable metric hybrid inexact proximal point method whose updating rule consists of an inexact proximal step and a projection step. Moreover, some specially designed stopping criteria for the inexact solution of the proximal subproblem are also used. However, due to the extra projection step, the connection between their algorithm and the proximal method of multipliers [42] is no longer available. Therefore, the results derived in [36] cannot be directly used to analyze the convergence properties of SNIPAL proposed in this paper, which is a variant of the proximal method of multipliers. We should also mention that in [17], Eckstein discussed nonlinear PPAs using Bregman functions, and the preconditioned PPA (1) may be viewed as a special instance if $\mathcal{M}_k$ is fixed for all k. However, the algorithms and convergence results in [17] are not applicable to our setting, where the linear operator $\mathcal{M}_k$ can change across iterations. More recently, the updating scheme (2.1) was also studied in [47, 48], where the authors presented various convergence results under the assumption that $\mathcal{P}_k(z^k)$ can be evaluated exactly for all $k \geq 0$. As one can observe later, this exact evaluation assumption is not suitable for our case. Since the scheme (2.1) under the classical setting of [41, 42] fits our context best, we conduct a comprehensive analysis of its convergence properties, which, to our best knowledge, are currently not available in the literature.

For all $k \geq 0$, define the mapping $\mathcal{Q}_k := \mathcal{I} - \mathcal{P}_k$. Clearly, if $0 \in \mathcal{T}(z)$, we have that $\mathcal{P}_k(z) = z$ and $\mathcal{Q}_k(z) = 0$ for all $k \geq 0$. Similar to [41, Proposition 1], we summarize the properties of $\mathcal{P}_k$ and $\mathcal{Q}_k$ in the following proposition.

PROPOSITION 2.2. *It holds for all $k \geq 0$ that*
(a) $z = \mathcal{P}_k(z) + \mathcal{Q}_k(z)$ *and* $c_k^{-1}\mathcal{M}_k\mathcal{Q}_k(z) \in \mathcal{T}(\mathcal{P}_k(z))$ *for all* $z \in \mathcal{X}$;
(b) $\langle \mathcal{P}_k(z) - \mathcal{P}_k(z'), \mathcal{Q}_k(z) - \mathcal{Q}_k(z') \rangle_{\mathcal{M}_k} \geq 0$ *for all* $z, z' \in \mathcal{X}$;
(c) $\|\mathcal{P}_k(z) - \mathcal{P}_k(z')\|^2_{\mathcal{M}_k} + \|\mathcal{Q}_k(z) - \mathcal{Q}_k(z')\|^2_{\mathcal{M}_k} \leq \|z - z'\|^2_{\mathcal{M}_k}$ *for all* $z, z' \in \mathcal{X}$.

Proof. The proof can be obtained via simple calculations and is similar to the proof of [41, Proposition 1]. We omit the details here. ∎

We list the following two general criteria for the approximate calculation of $\mathcal{P}_k(z^k)$, which are analogous to those proposed in [41]:

(A) $\|z^{k+1} - \mathcal{P}_k(z^k)\|_{\mathcal{M}_k} \leq \epsilon_k, \quad 0 \leq \epsilon_k, \quad \sum_{k=0}^{\infty} \epsilon_k < \infty,$

(B) $\|z^{k+1} - \mathcal{P}_k(z^k)\|_{\mathcal{M}_k} \leq \delta_k \|z^{k+1} - z^k\|_{\mathcal{M}_k}, \quad 0 \leq \delta_k < 1, \quad \sum_{k=0}^{\infty} \delta_k < \infty.$

THEOREM 2.3. *Suppose that $\Omega = \mathcal{T}^{-1}(0) \neq \emptyset$. Let $\{z^k\}$ be any sequence generated by the mPPA* (2.1) *under criterion* (A). *Then $\{z^k\}$ is bounded and*

$$\text{dist}_{\mathcal{M}_{k+1}}(z^{k+1}, \Omega) \leq (1 + \nu_k)\text{dist}_{\mathcal{M}_k}(z^k, \Omega) + (1 + \nu_k)\epsilon_k \quad \forall k \geq 0. \tag{2.4}$$

In addition, $\{z^k\}$ converges to a point z^∞ such that $0 \in \mathcal{T}(z^\infty)$.

Proof. Let $\bar{z} \in \mathcal{X}$ be a point satisfying $0 \in \mathcal{T}(\bar{z})$. It is readily shown that $\bar{z} = \mathcal{P}_k(\bar{z})$. We have

$$\|z^{k+1} - \bar{z}\|_{\mathcal{M}_k} - \epsilon_k \leq \|\mathcal{P}_k(z^k) - \bar{z}\|_{\mathcal{M}_k} = \|\mathcal{P}_k(z^k) - \mathcal{P}_k(\bar{z})\|_{\mathcal{M}_k} \leq \|z^k - \bar{z}\|_{\mathcal{M}_k}. \tag{2.5}$$

Since $(1+\nu_k)\mathcal{M}_k \succeq \mathcal{M}_{k+1}$, we know that

$$\|z^{k+1}-\bar{z}\|_{\mathcal{M}_{k+1}} \le (1+\nu_k)\|z^{k+1}-\bar{z}\|_{\mathcal{M}_k} \le (1+\nu_k)\|z^k-\bar{z}\|_{\mathcal{M}_k} + (1+\nu_k)\epsilon_k. \tag{2.6}$$

Let $\Pi_\Omega(z)$ denote the projection of z onto Ω. By noting that $0 \in \mathcal{T}(\Pi_\Omega(z^k))$, we get from the above inequality (by setting $\bar{z} = \Pi_\Omega(z^k)$) that

$$\begin{aligned}\mathrm{dist}_{\mathcal{M}_{k+1}}(z^{k+1},\Omega) &\le \|z^{k+1}-\Pi_\Omega(z^k)\|_{\mathcal{M}_{k+1}} \\ &\le (1+\nu_k)\|z^k-\Pi_\Omega(z^k)\|_{\mathcal{M}_k} + (1+\nu_k)\epsilon_k \\ &= (1+\nu_k)\mathrm{dist}_{\mathcal{M}_k}(z^k,\Omega) + (1+\nu_k)\epsilon_k.\end{aligned}$$

Since

$$\sum_{k=0}^{\infty}(1+\nu_k)\epsilon_k \le \sum_{k=0}^{\infty}\epsilon_k + (\max_{k\ge 0}\epsilon_k)\sum_{k=0}^{\infty}\nu_k < +\infty,$$

we know from [35, Lemma 2.2.2], (2.5), and (2.6) that

$$\lim_{k\to\infty}\|z^k-\bar{z}\|_{\mathcal{M}_k} = \lim_{k\to\infty}\|z^{k+1}-\bar{z}\|_{\mathcal{M}_k} = \mu < \infty \quad \text{and} \quad \lim_{k\to\infty}\|\mathcal{P}_k(z^k)-\bar{z}\|_{\mathcal{M}_k} = \mu. \tag{2.7}$$

The boundedness of $\{z^k\}$ thus follows directly from the fact that $\mathcal{M}_k \succeq \lambda_{\min}\mathcal{I}$ for all $k \ge 0$. Therefore, $\{z^k\}$ has at least one cluster point z^∞.

From Proposition 2.2, we know that for all $k \ge 0$

$$0 \le \|\mathcal{Q}_k(z^k)\|^2_{\mathcal{M}_k} \le \|z^k-\bar{z}\|^2_{\mathcal{M}_k} - \|\mathcal{P}_k(z^k)-\bar{z}\|^2_{\mathcal{M}_k}. \tag{2.8}$$

Therefore, $\lim_{k\to\infty}\|\mathcal{Q}_k(z^k)\|^2_{\mathcal{M}_k} = 0$. It follows that

$$\lim_{k\to\infty} c_k^{-1}\mathcal{M}_k\mathcal{Q}_k(z^k) = \lim_{k\to\infty}\mathcal{Q}_k(z^k) = 0, \tag{2.9}$$

because the number c_k is bounded away from zero and $\mathcal{M}_k \succeq \lambda_{\min}\mathcal{I}$ for all $k \ge 0$. Since

$$\|\mathcal{Q}_k(z^k)\|_{\mathcal{M}_k} = \|(z^k-z^{k+1})+(z^{k+1}-\mathcal{P}_k(z^k))\|_{\mathcal{M}_k} \ge \|z^k-z^{k+1}\|_{\mathcal{M}_k} - \epsilon_k,$$

we further have $\lim_{k\to\infty}\|z^k-z^{k+1}\| = 0$.

Since z^∞ is a cluster point of z^k and

$$\lim_{k\to\infty}\|\mathcal{P}_k(z^k)-z^{k+1}\| = \lim_{k\to\infty}\|z^{k+1}-z^k\| = 0,$$

z^∞ is also a cluster point of $\mathcal{P}_k(z^k)$. From Proposition 2.2(a), we have that for any $w \in \mathcal{T}(z)$

$$0 \le \langle z-\mathcal{P}_k(z^k),\, w - c_k^{-1}\mathcal{M}_k\mathcal{Q}_k(z^k)\rangle \quad \forall\, k \ge 0,$$

which, together with (2.9), implies

$$0 \le \langle z-z^\infty,\, w\rangle \quad \forall\, z,w \text{ satisfying } w \in \mathcal{T}(z).$$

From the maximality of $\mathcal{T}$, we know that $0 \in \mathcal{T}(z^\infty)$. Hence, we can replace $\bar{z}$ in (2.7) by z^∞. Therefore,

$$\lim_{k\to\infty}\|z^k-z^\infty\|_{\mathcal{M}_k} = 0.$$

That is $\lim_{k\to\infty} z^k = z^\infty$. ∎

Next, we study the convergence rate of the preconditioned PPA. The following error bound assumption associated with $\mathcal{T}$ is critical to the study of the convergence rate of the preconditioned PPA.

ASSUMPTION 2. *For any $r > 0$, there exists $\kappa > 0$ such that*

$$\text{dist}(x, \mathcal{T}^{-1}(0)) \le \kappa\, \text{dist}(0, \mathcal{T}(x)) \quad \forall\, x \in \mathcal{X} \text{ satisfying } \text{dist}(x, \mathcal{T}^{-1}(0)) \le r. \tag{2.10}$$

In Rockafellar's classic work [41], the asymptotic Q-superlinear convergence of PPA is established under the assumption that $\mathcal{T}^{-1}$ is Lipschitz continuous at zero. Note that the Lipschitz continuity assumption on $\mathcal{T}^{-1}$ is rather restrictive, since it implicitly implies that $\mathcal{T}^{-1}(0)$ is a singleton. In [29], Luque extended Rockafellar's work by considering the following relaxed condition over $\mathcal{T}$: there exist $\gamma > 0$ and $\epsilon > 0$ such that

$$\text{dist}(x, \mathcal{T}^{-1}(0)) \le \gamma \text{dist}(0, \mathcal{T}(x)) \quad \forall\, x \in \{x \in \mathcal{X} \mid \text{dist}(0, \mathcal{T}(x)) < \epsilon\}. \tag{2.11}$$

We show in the following lemma that this condition in fact implies Assumption 2. Thus, our Assumption 2 is quite mild and weaker than condition (2.11).

LEMMA 2.4. *Let F be a multifunction from $\mathcal{X}$ to $\mathcal{Y}$ with $F^{-1}(0) \neq \emptyset$. If F satisfies condition* (2.11)*, then Assumption* 2 *holds for F, i.e., for any $r > 0$, there exists $\kappa > 0$ such that*

$$\text{dist}(x, F^{-1}(0)) \le \kappa\, \text{dist}(0, F(x)) \quad \forall\, x \in \mathcal{X} \text{ satisfying } \text{dist}(x, F^{-1}(0)) \le r.$$

Proof. Since F satisfies condition (2.11), there exist $\varepsilon > 0$ and $\kappa_0 \ge 0$ such that if $x \in \mathcal{X}$ satisfies $\text{dist}(0, F(x)) < \varepsilon$, then

$$\text{dist}(x, F^{-1}(0)) \le \kappa_0 \text{dist}(0, F(x)).$$

For any $r > 0$ and x satisfying $\text{dist}(x, F^{-1}(0)) \le r$, if $\text{dist}(0, F(x)) < \epsilon$, then $\text{dist}(x, F^{-1}(0)) \le \kappa_0 \text{dist}(0, F(x))$; otherwise if $\text{dist}(0, F(x)) \ge \epsilon$, then

$$\text{dist}(0, F(x)) \ge \epsilon \ge \frac{\epsilon}{r} \text{dist}(x, F^{-1}(0)),$$

i.e., $\text{dist}(x, F^{-1}(0)) \le \frac{r}{\epsilon} \text{dist}(0, F(x))$. Therefore, the desired inequality holds for $\kappa = \max\{\kappa_0, \frac{r}{\epsilon}\}$. ∎

Remark 1. In fact, condition (2.11) is exactly the local upper Lipschitz continuity of $\mathcal{T}^{-1}$ at the origin, which was introduced by Robinson in [38]. Later, Robinson established in [39] the celebrated result that every polyhedral multifunction is locally upper Lipschitz continuous, i.e., satisfies condition (2.11). Thus from Lemma 2.4, we know that any polyhedral multifunction F with $F^{-1}(0) \neq \emptyset$ satisfies Assumption 2. We note that Assumption 2 is also employed and studied in [53].

Since the nonnegative sequences $\{\nu_k\}$ and $\{\epsilon_k\}$ in condition (2.2) and the stopping criterion (A), respectively, are summable, we know that $0 < \Pi_{k=0}^{\infty}(1 + \nu_k) < +\infty$ and we can choose r to be a positive number satisfying $r > \sum_{k=0}^{\infty} \epsilon_k (1 + \nu_k)$. Assume that $\mathcal{T}$ satisfies Assumption 2; then associated with r, there exists a positive constant κ such that (2.10) holds. With these preparations, we prove in the following theorem the asymptotic Q-superlinear (R-superlinear) convergence of the weighted (unweighted) distance between the sequence generated by the preconditioned PPA and Ω.

THEOREM 2.5. *Suppose that $\Omega \neq \emptyset$ and the initial point z^0 satisfies*

$$\mathrm{dist}_{\mathcal{M}_0}(z^0, \Omega) \leq \frac{r - \sum_{k=0}^{\infty} \epsilon_k (1+\nu_k)}{\Pi_{k=0}^{\infty}(1+\nu_k)}.$$

Let $\{z^k\}$ be the infinite sequence generated by the preconditioned PPA under criteria (A) *and* (B) *with $\{c_k\}$ nondecreasing ($c_k \uparrow c_\infty \leq \infty$). Then for all $k \geq 0$, it holds that*

$$\mathrm{dist}_{\mathcal{M}_{k+1}}(z^{k+1}, \Omega) \leq \mu_k \mathrm{dist}_{\mathcal{M}_k}(z^k, \Omega), \tag{2.12}$$

where $\mu_k = (1+\nu_k)(1-\delta_k)^{-1}\big(\delta_k + (1+\delta_k)\kappa\lambda_{\max}(\mathcal{M}_k)/\sqrt{c_k^2 + \kappa^2\lambda_{\max}^2(\mathcal{M}_k)}\,\big)$ and

$$\limsup_{k\to\infty} \mu_k = \mu_\infty = \frac{\kappa\lambda_\infty}{\sqrt{c_\infty^2 + \kappa^2\lambda_\infty^2}} < 1 \quad (\mu_\infty = 0 \text{ if } c_\infty = \infty) \tag{2.13}$$

with λ_∞ given in (2.2). *In addition, one has that for all $k \geq 0$,*

$$\mathrm{dist}(z^{k+1}, \Omega) \leq \frac{\mu_k}{\sqrt{\lambda_{\min}(\mathcal{M}_{k+1})}} \mathrm{dist}_{\mathcal{M}_k}(z^k, \Omega). \tag{2.14}$$

Proof. From (2.4) in Theorem 2.3, we know that for all $k \geq 0$, $\mathrm{dist}_{\mathcal{M}_k}(z^k, \Omega) \leq \Pi_{k=0}^{\infty}(1+\nu_k)\mathrm{dist}_{\mathcal{M}_0}(z^0, \Omega) + \sum_{k=0}^{\infty}\epsilon_k(1+\nu_k) \leq r$, and consequently,

$$\begin{aligned}\mathrm{dist}_{\mathcal{M}_k}(\mathcal{P}_k(z^k), \Omega) &\leq \|\mathcal{P}_k(z^k) - \Pi_\Omega(z^k)\|_{\mathcal{M}_k} \\ &= \|\mathcal{P}_k(z^k) - \mathcal{P}_k(\Pi_\Omega(z^k))\|_{\mathcal{M}_k} \leq \mathrm{dist}_{\mathcal{M}_k}(z^k, \Omega) \leq r \quad \forall k \geq 0.\end{aligned}$$

From Proposition 2.2(a), we have

$$c_k^{-1}\mathcal{M}_k\mathcal{Q}_k(z^k) \in \mathcal{T}(\mathcal{P}_k(z^k)),$$

which, together with Assumption 2, implies that for all $k \geq 0$

$$\mathrm{dist}(\mathcal{P}_k(z^k), \Omega) \leq \kappa c_k^{-1}\|\mathcal{M}_k\mathcal{Q}_k(z^k)\|.$$

It further implies that for all $k \geq 0$,

$$\frac{1}{\sqrt{\lambda_{\max}(\mathcal{M}_k)}}\mathrm{dist}_{\mathcal{M}_k}(\mathcal{P}_k(z^k), \Omega) \leq \mathrm{dist}(\mathcal{P}_k(z^k), \Omega) \leq \sqrt{\lambda_{\max}(\mathcal{M}_k)}\kappa c_k^{-1}\|\mathcal{Q}_k(z^k)\|_{\mathcal{M}_k}.$$

Now taking $\bar{z} = \Pi_\Omega(z^k)$, we deduce from (2.8) that for all $k \geq 0$,

$$\begin{aligned}\|\mathcal{Q}_k(z^k)\|_{\mathcal{M}_k}^2 &\leq \|z^k - \Pi_\Omega(z^k)\|_{\mathcal{M}_k}^2 - \|\mathcal{P}_k(z^k) - \Pi_\Omega(z^k)\|_{\mathcal{M}_k}^2 \\ &\leq \mathrm{dist}_{\mathcal{M}_k}^2(z^k, \Omega) - \mathrm{dist}_{\mathcal{M}_k}^2(\mathcal{P}_k(z^k), \Omega).\end{aligned} \tag{2.15}$$

Therefore, it holds that

$$\mathrm{dist}_{\mathcal{M}_k}(\mathcal{P}_k(z^k), \Omega) \leq \frac{\kappa\lambda_{\max}(\mathcal{M}_k)}{\sqrt{c_k^2 + \kappa^2\lambda_{\max}^2(\mathcal{M}_k)}}\mathrm{dist}_{\mathcal{M}_k}(z^k, \Omega) \quad \forall\, k \geq 0. \tag{2.16}$$

Under stopping criterion (B), we further have for all $k \geq 0$,

$$\begin{aligned}&\|z^{k+1} - \Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k} \\ &\quad\leq \|z^{k+1} - \mathcal{P}_k(z^k)\|_{\mathcal{M}_k} + \|\mathcal{P}_k(z^k) - \Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k} \\ &\quad\leq \delta_k\|z^{k+1} - z^k\|_{\mathcal{M}_k} + \|\mathcal{P}_k(z^k) - \Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k} \\ &\quad\leq \delta_k\big(\|z^{k+1} - \Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k} + \|z^k - \Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}\big) \\ &\qquad + \|\mathcal{P}_k(z^k) - \Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}.\end{aligned}$$

Thus,

$$\begin{aligned}&(1-\delta_k)\|z^{k+1}-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}\\&\quad\le \delta_k\|z^k-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}+\|\mathcal{P}_k(z^k)-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}.\end{aligned}$$

Now

$$\begin{aligned}&\delta_k\|z^k-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}\\&\quad\le \delta_k\|\mathcal{P}_k(z^k)-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}+\delta_k\|\mathcal{Q}_k(z^k)\|_{\mathcal{M}_k}\\&\quad\le \delta_k\|\mathcal{P}_k(z^k)-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}+\delta_k\mathrm{dist}_{\mathcal{M}_k}(z^k,\Omega),\end{aligned}$$

where the last inequality follows from (2.15). By using the above inequality in the previous one, we get

$$(1-\delta_k)\|z^{k+1}-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}\ \le\ \delta_k\mathrm{dist}_{\mathcal{M}_k}(z^k,\Omega)+(1+\delta_k)\mathrm{dist}_{\mathcal{M}_k}(\mathcal{P}_k(z^k),\Omega).$$

Therefore, from the last inequality and (2.16), it holds that for all $k\ge 0$,

$$\begin{aligned}\mathrm{dist}_{\mathcal{M}_{k+1}}(z^{k+1},\Omega)\le&(1+\nu_k)\mathrm{dist}_{\mathcal{M}_k}(z^{k+1},\Omega)\\\le&(1+\nu_k)\|z^{k+1}-\Pi_\Omega(\mathcal{P}_k(z^k))\|_{\mathcal{M}_k}\le\mu_k\mathrm{dist}_{\mathcal{M}_k}(z^k,\Omega),\end{aligned}$$

where $\mu_k=(1+\nu_k)(1-\delta_k)^{-1}(\delta_k+(1+\delta_k)\kappa\lambda_{\max}(\mathcal{M}_k)/\sqrt{c_k^2+\kappa^2\lambda^2_{\max}(\mathcal{M}_k)})$. That is, (2.12) holds for all $k\ge 0$. Since for all $k\ge 0$, $\mathcal{M}_k\succeq\lambda_{\min}\mathcal{I}$, (2.13) and (2.14) can be obtained through simple calculations. ∎

Remark 2. In the theorem, the assumption on the initial point z^0 is inspired by the similar one assumed in [53, Lemma 4.1]. Suppose that $\{\delta_k\}$ in criterion (B) is nonincreasing and $\nu_k\equiv 0$ for all $k\ge 0$. Since $\{c_k\}$ is nondecreasing and $\lambda_{\max}(\mathcal{M}_k)$ is nonincreasing, we know that $\{\mu_k\}$ is nonincrasing. Therefore, if one chooses c_0 large enough such that $\mu_0<1$, then we have $\mu_k\le\mu_0<1$ for all $k\ge 0$. The inequality (2.12) thus implies the global Q-linear convergence of $\{\mathrm{dist}_{\mathcal{M}_k}(z^k,\Omega)\}$. In addition, (2.14) implies that for all $k\ge 0$,

$$\mathrm{dist}(z^{k+1},\Omega)\le\big(\mathrm{dist}_{\mathcal{M}_0}(z^0,\Omega)/\sqrt{\lambda_{\min}}\big)\Pi_{i=0}^k\mu_i\le(\mu_0)^{k+1}\big(\mathrm{dist}_{\mathcal{M}_0}(z^0,\Omega)/\sqrt{\lambda_{\min}}\big),$$

i.e., $\{\mathrm{dist}(z^k,\Omega)\}$ converges globally R-linearly.

3. A semismooth Newton proximal augmented Lagragian method. Note that we can equivalently rewrite problem (D) in the following minimization form:

$$\text{(D)}\quad -\min\Big\{g(y):=\delta_K^*(A^*y-c)-b^Ty\Big\}.$$

Associated with this unconstrained formulation, we write the augmented Lagrangian function following the framework developed in [43, Examples 11.46 and 11.57]. To do so, we first identify (D) with the problem of minimizing $g(y)=\widetilde{g}(y,0)$ over $\mathbb{R}^m$ for

$$\widetilde{g}(y,\xi)=-b^Ty+\delta_K^*(A^*y-c+\xi)\quad\forall\,(y,\xi)\in\mathbb{R}^m\times\mathbb{R}^n.$$

Obviously, $\widetilde{g}$ is jointly convex in (y,ξ). Now, we are able to write down the Lagrangian function $l:\mathbb{R}^m\times\mathbb{R}^n$ through partial dualization as follows:

$$l(y;x):=\inf_\xi\{\widetilde{g}(y,\xi)-\langle x,\,\xi\rangle\}=-b^Ty-\langle x,\,c-A^*y\rangle-\delta_K(x).$$

Thus, the KKT conditions associated with (P) and (D) are given by

$$-b + Ax = 0, \quad A^*y - c \in \partial\delta_K(x), \quad (x, y) \in \mathbb{R}^n \times \mathbb{R}^m. \tag{3.1}$$

Given $\sigma > 0$, the augmented Lagrangian function corresponding to (D) can be obtained by

$$\begin{aligned} L_\sigma(y; x) :=& \sup_{s\in\mathbb{R}^n} \Big\{ l(y; s) - \frac{1}{2\sigma}\|s - x\|^2 \Big\} \\ =& -b^T y - \inf_{s\in\mathbb{R}^n} \Big\{ \delta_K(s) + \langle s,\, c - A^*y\rangle + \frac{1}{2\sigma}\|s - x\|^2 \Big\} \\ =& -b^T y - \langle \Pi_K(x - \sigma(c - A^*y)),\, c - A^*y\rangle - \frac{1}{2\sigma}\|\Pi_K(x - \sigma(c - A^*y)) - x\|^2. \end{aligned}$$

We propose to solve (D) via an inexact proximal augmented Lagrangian method. Our algorithm is named the semismooth Newton inexact proximal augmented Lagrangian (SNIPAL) method because we will design a semismooth Newton method to solve the underlying augmented Lagrangian subproblems. Its template is given as follows.

Algorithm 3.1. SNIPAL: Semismooth Newton inexact proximal augmented Lagrangian.

Let $\sigma_0, \sigma_\infty > 0$ be given parameters, $\{\tau_k\}_{k=0}^\infty$ be a given nonincreasing sequence such that $\tau_k > 0$ for all $k \geq 0$. Choose $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m$. For $k = 1, \ldots,$ perform the following steps in each iteration.

Step 1. Compute

$$y^{k+1} \approx \operatorname{argmin}_{y\in\mathbb{R}^m} \Big\{ L_{\sigma_k}(y; x^k) + \frac{\tau_k}{2\sigma_k}\|y - y^k\|^2 \Big\} \tag{3.2}$$

via the semismooth Newton method.

Step 2. Compute $x^{k+1} = \Pi_K\big(x^k - \sigma_k(c - A^*y^{k+1})\big)$.

Step 3. Update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$.

Note that unlike the case in the classic proximal method of multipliers in [42] with $\tau_k \equiv 1$ for all k, we allow an adaptive choice of the parameter τ_k in the proximal term $\frac{\tau_k}{2\sigma_k}\|y - y^k\|^2$ in the inner subproblem (3.2) of the SNIPAL algorithm. Here, the proximal term is added to guarantee the existence of the optimal solution to the inner subproblem (3.2) and to ensure the positive definiteness of the coefficient matrix of the underlying semismooth Newton linear system. Moreover, our numerical experience with SNIPAL indicates that having the additional flexibility of choosing the parameter τ_k can help to improve the practical performance of the algorithm. We emphasize here that comparing to [42], our modifications focus more on the computational and implementational aspects.

While the introduction of the parameters $\{\tau_k\}$ brings us more flexibility and some promising numerical advantages, it also makes the convergence analysis of the algorithm more challenging. Fortunately, we are able to rigorously characterize the connection between our SNIPAL Algorithm and the preconditioned PPA studied in section 2. As one will see in the subsequent text, this connection allows us to conduct a comprehensive convergence analysis for the SNIPAL Algorithm. From the convergence analysis, we also note that $\frac{\tau_k}{2\sigma_k}\|y - y^k\|^2$ can be replaced by a more general proximal term, i.e., $\frac{1}{2\sigma_k}\|y - y^k\|_{T_k}^2$ with a symmetric positive definite matrix T_k.

3.1. Global convergence properties of SNIPAL. In this section, we present a comprehensive analysis for the convergence properties of SNIPAL. The global convergence and global linear-rate convergence of SNIPAL are presented as an application of the theory of the preconditioned PPA.

To establish the connection between SNIPAL and the preconditioned PPA, we first introduce some notation. To this end, for $k = 0, 1, \ldots$ and any given $(\bar{y}, \bar{x}) \in \mathbb{R}^m \times \mathbb{R}^n$, define the function

$$P_k(\bar{y}, \bar{x}) := \arg\operatorname*{minimax}_{y,x} \Big\{ l(y, x) + \frac{\tau_k}{2\sigma_k}\|y - \bar{y}\|^2 - \frac{1}{2\sigma_k}\|x - \bar{x}\|^2 \Big\}. \tag{3.3}$$

Corresponding to the closed proper convex-concave function l, we can define the maximal monotone operator $\mathcal{T}_l$ [40, Corollary 37.5.2], by

$$\begin{aligned} \mathcal{T}_l(y, x) &:= \{(y', x') \,|\, (y', -x') \in \partial l(y, x)\} \\ &= \{(y', x') \,|\, y' = -b + Ax,\ x' \in c - A^*y + \partial\delta_K(x)\}, \end{aligned}$$

whose corresponding inverse operator is given by

$$\mathcal{T}_l^{-1}(y', x') := \arg\operatorname*{minimax}_{y,x} \{l(y, x) - \langle y', y\rangle + \langle x', x\rangle\}. \tag{3.4}$$

Since K is a polyhedral set, $\partial\delta_K$ is known to be a polyhedral multifunction (see, e.g., [27, p. 108]). As the sum of two polyhedral multifunctions is also polyhedral, $\mathcal{T}_l$ is also polyhedral. Define, for $k = 0, 1, \ldots,$

$$\Lambda_k = \mathrm{Diag}\,(\tau_k I_m, I_n) \succ 0. \tag{3.5}$$

The optimal solution of problem (3.3), i.e., $P_k(\bar{y}, \bar{x})$, can be obtained via the following lemma.

LEMMA 3.1. *For all $k \geq 0$, it holds that*

$$P_k(\bar{y}, \bar{x}) = (\Lambda_k + \sigma_k\mathcal{T}_l)^{-1}\Lambda_k(\bar{y}, \bar{x}) \quad \forall\, (\bar{y}, \bar{x}) \in \mathbb{R}^m \times \mathbb{R}^n. \tag{3.6}$$

If $(y^*, x^*) \in \mathcal{T}_l^{-1}(0)$, then $P_k(y^*, x^*) = (y^*, x^*)$.

In SNIPAL, at the kth iteration, denote

$$\psi_k(y) := L_{\sigma_k}(y; x^k) + \frac{\tau_k}{2\sigma_k}\|y - y^k\|^2. \tag{3.7}$$

From the property of the proximal mapping, we know that ψ_k is continuously differentiable and

$$\nabla\psi_k(y) = -b + A\,\Pi_K\big(x^k + \sigma_k(A^*y - c)\big) + \tau_k\sigma_k^{-1}(y - y^k).$$

As a generalization of Proposition 8 in [42], the following proposition about the weighted distance between (y^{k+1}, x^{k+1}) generated by SNIPAL and $P_k(y^k, x^k)$ is important for designing the stopping criteria for the subproblem (3.2) and establishing the connection between SNIPAL and the preconditioned PPA.

PROPOSITION 3.2. *Let P_k, Λ_k, and ψ_k be defined in* (3.3), (3.5), *and* (3.7), *respectively. Let (y^{k+1}, x^{k+1}) be generated by the* SNIPAL *algorithm at iteration $k+1$. It holds that*

$$\|(y^{k+1}, x^{k+1}) - P_k(y^k, x^k)\|_{\Lambda_k} \leq \frac{\sigma_k}{\min(\sqrt{\tau_k}, 1)}\,\|\nabla\psi_k(y^{k+1})\|. \tag{3.8}$$

Proof. Since $\nabla\psi_k(y^{k+1}) = \nabla_y L_{\sigma_k}(y^{k+1}, x^k) + \tau_k\sigma_k^{-1}(y^{k+1} - y^k)$, we have

$$\nabla\psi_k(y^{k+1}) + \sigma_k^{-1}\tau_k(y^k - y^{k+1}) = \nabla_y L_{\sigma_k}(y^{k+1}, x^k),$$

which, by [42, Proposition 7], implies $(\nabla\psi_k(y^{k+1})+\sigma_k^{-1}\tau_k(y^k-y^{k+1}), \sigma_k^{-1}(x^k-x^{k+1})) \in \mathcal{T}_l(y^{k+1}, x^{k+1})$. Thus,

$$\sigma_k(\nabla\psi_k(y^{k+1}), 0) + \Lambda_k\big((y^k, x^k) - (y^{k+1}, x^{k+1})\big) \in \sigma_k\mathcal{T}_l(y^{k+1}, x^{k+1})$$

and $\sigma_k(\nabla\psi_k(y^{k+1}), 0) + \Lambda_k(y^k, x^k) \in (\Lambda_k + \sigma_k\mathcal{T}_l)(y^{k+1}, x^{k+1})$, or equivalently,

$$(y^{k+1}, x^{k+1}) = (\Lambda_k + \sigma_k\mathcal{T}_l)^{-1}\Lambda_k\big(\Lambda_k^{-1}(\sigma_k\nabla\psi_k(y^{k+1}), 0) + (y^k, x^k)\big).$$

Then, by Lemma 3.1 and Proposition 2.2, we know that

$$\begin{aligned}
&\|(y^{k+1}, x^{k+1}) - P_k(y^k, x^k)\|_{\Lambda_k} \\
&\quad = \|(\Lambda_k + \sigma_k\mathcal{T}_l)^{-1}\Lambda_k\big(\Lambda_k^{-1}(\sigma_k\nabla\psi_k(y^{k+1}), 0) + (y^k, x^k)\big) \\
&\qquad - (\Lambda_k + \sigma_k\mathcal{T}_l)^{-1}\Lambda_k\big((y^k, x^k)\big)\|_{\Lambda_k} \\
&\quad \le \|\Lambda_k^{-1}\big(\sigma_k\nabla\psi_k(y^{k+1}), 0\big)\|_{\Lambda_k} \le \frac{\sigma_k}{\min\,(\sqrt{\tau_k}, 1)}\|\nabla\psi_k(y^{k+1})\|.
\end{aligned}$$

This completes the proof for the proposition. ∎

Based on Proposition 3.2, we propose the following stopping criteria for the approximate computation of y^{k+1} in Step 1 of SNIPAL:

$$(\mathrm{A}')\quad \|\nabla\psi_k(y^{k+1})\| \le \frac{\min(\sqrt{\tau_k}, 1)}{\sigma_k}\epsilon_k, \quad 0 \le \epsilon_k, \quad \sum_{k=0}^{\infty}\epsilon_k < \infty,$$

$$(\mathrm{B}')\quad \|\nabla\psi_k(y^{k+1})\| \le \frac{\delta_k\min(\sqrt{\tau_k}, 1)}{\sigma_k}\|(y^{k+1}, x^{k+1}) - (y^k, x^k)\|_{\Lambda_k}, \quad 0 \le \delta_k < 1,\ \sum_{k=0}^{\infty}\delta_k < \infty.$$

For the convergence of SNIPAL, we also need the following assumption on τ_k.

ASSUMPTION 3. *The positive sequence $\{\tau_k\}$ is nonincreasing and bounded away from zero, i.e., $\tau_k \downarrow \tau_\infty > 0$ for some positive constant τ_∞.*

Under Assumption 3, we have that for all $k \ge 0$,

$$\Lambda_k \succeq \Lambda_{k+1} \text{ and } \Lambda_k \succeq \min(1, \tau_\infty)I_{m+n}.$$

We now present the global convergence result for SNIPAL in the following theorem. Similar to the case in [42], it is in fact a direct application of Theorem 2.3.

THEOREM 3.3 (global convergence of SNIPAL). *Suppose that Assumptions* 1 *and* 3 *hold. Let $\{(y^k, x^k)\}$ be the sequence generated by the* SNIPAL *algorithm with the stopping criterion* (A′). *Then $\{(y^k, x^k)\}$ is bounded. In addition, $\{x^k\}$ converges to an optimal solution of* (P) *and $\{y^k\}$ converges to an optimal solution of* (D), *respectively.*

Since $\mathcal{T}_l$ is a polyhedral multifunction, we know from Lemma 2.4 and Remark 1 that $\mathcal{T}_l$ satisfies Assumption 2. Let r be a positive number satisfying $r > \sum_{i=0}^{\infty} \epsilon_k$ with ϵ_k being the summable sequence in (A'). Then, there exists $\kappa > 0$ associated with r such that for any $(y, x) \in \mathbb{R}^m \times \mathbb{R}^n$ satisfying $\mathrm{dist}((y,x), \mathcal{T}_l^{-1}(0)) \le r$,

$$\mathrm{dist}((y,x), \mathcal{T}_l^{-1}(0)) \le \kappa\, \mathrm{dist}(0, \mathcal{T}_l(y,x)). \tag{3.9}$$

As an application of Theorem 2.5, we are now ready to show the asymptotic superlinear convergence of SNIPAL in the following theorem.

THEOREM 3.4 (asymptotic superlinear convergence of SNIPAL). *Suppose that Assumptions* 1 *and* 3 *hold and the initial* $z^0 := (y^0, x^0)$ *satisfies* $\mathrm{dist}_{\Lambda_0}(z^0, \mathcal{T}_l^{-1}(0)) \le r - \sum_{i=0}^{\infty} \epsilon_k$. *Let* κ *be the modulus given in* (3.9) *and* $\{z^k := (y^k, x^k)\}$ *be the infinite sequence generated by the preconditioned PPA under criteria* (A′) *and* (B′). *Then, for all* $k \ge 0$, *it holds that*

$$\begin{aligned} &\mathrm{dist}_{\Lambda_{k+1}}(z^{k+1}, \mathcal{T}_l^{-1}(0)) \le \mu_k \mathrm{dist}_{\Lambda_k}(z^k, \mathcal{T}_l^{-1}(0)), \\ &\mathrm{dist}(z^{k+1}, \mathcal{T}_l^{-1}(0)) \le \frac{\mu_k}{\sqrt{\min(1, \tau_{k+1})}} \mathrm{dist}_{\Lambda_k}(z^k, \mathcal{T}_l^{-1}(0)), \end{aligned} \tag{3.10}$$

where $\mu_k = (1-\delta_k)^{-1}\Big(\delta_k + (1+\delta_k)\kappa\gamma_k / \sqrt{\sigma_k^2 + \kappa^2\gamma_k^2}\Big)$ *with* $\gamma_k := \max(\tau_k, 1)$ *and*

$$\lim_{k\to\infty} \mu_k = \mu_\infty = \frac{\kappa\gamma_\infty}{\sqrt{\sigma_\infty^2 + \kappa^2\gamma_\infty^2}} < 1 \quad (\mu_\infty = 0 \text{ if } \sigma_\infty = \infty)$$

with $\gamma_\infty = \max(\tau_\infty, 1)$.

Remark 3. Suppose that $\{\delta_k\}$ in criterion (B′) is nonincreasing. We know from Remark 2 that if one chooses σ_0 large enough such that $\mu_0 < 1$, then $\mu_k \le \mu_0 < 1$ for all $k \ge 0$. Thus, from (3.10), we have the global linear convergence of $\{\mathrm{dist}_{\Lambda_k}(z^k, \mathcal{T}_l^{-1}(0))\}$ and $\{\mathrm{dist}(z^k, \mathcal{T}_l^{-1}(0))\}$.

3.2. Semismooth Newton method for subproblems (3.2). In this subsection, we discuss how the subproblem (3.2) in SNIPAL can be solved efficiently. As is mentioned in the name of SNIPAL, we propose to solve (3.2) via an inexact semismooth Newton method, which converges at least locally superlinearly. In fact, the local convergence rate can even be quadratic.

For given $(\tilde{x}, \tilde{y}) \in \mathbb{R}^n \times \mathbb{R}^m$ and $\tau, \sigma > 0$, define the function $\psi : \mathbb{R}^m \to \mathbb{R}$ as

$$\psi(y) := L_\sigma(y; \tilde{x}) + \frac{\tau}{2\sigma}\|y - \tilde{y}\|^2 \quad \forall y \in \mathbb{R}^m,$$

and we aim to solve

$$\min_{y\in\mathbb{R}^m} \psi(y). \tag{3.11}$$

Note that ψ is strongly convex and continuously differentiable over $\mathbb{R}^m$ with

$$\nabla\psi(y) = -b + A\,\Pi_K\big(\tilde{x} + \sigma(A^*y - c)\big) + \tau\sigma^{-1}(y - \tilde{y}).$$

Hence, we know that for any given $\alpha \ge \inf_y \psi(y)$, the level set $\mathcal{L}_\alpha := \{y \in \mathbb{R}^m \mid \psi(y) \le \alpha\}$ is a nonempty closed and bounded convex set. In addition, problem (3.11) has a unique optimal solution which we denote as $\bar{y}$.

As an unconstrained optimization problem, the optimality condition for (3.11) is given by

$$\nabla\psi(y) = 0, \quad y \in \mathbb{R}^m, \tag{3.12}$$

and $\bar{y}$ is the unique solution to this nonsmooth equation. Since Π_K is a Lipschitz continuous piecewise affine function, we have that $\nabla\psi$ is strongly semismooth. Hence, we can solve the nonsmooth equation (3.12) via a semismooth Newton method. For this purpose, we define the following operator:

$$\hat{\partial}^2\psi(y) := \tau\sigma^{-1}I_m + \sigma A\partial\Pi_K(\tilde{x} + \sigma(A^*y - c))A^* \quad \forall y \in \mathbb{R}^m,$$

where $\partial\Pi_K(\tilde{x}+\sigma(A^*y-c))$ is the Clarke subdifferential [14] of the Lipschitz continuous mapping $\Pi_K(\cdot)$ at $\tilde{x} + \sigma(A^*y - c)$. Note that from [25, Example 2.5], we have that

$$\hat{\partial}^2\psi(y)d = \partial^2\psi(y)d \quad \forall d \in \mathbb{R}^m,$$

where $\partial^2\psi(y)$ denotes the generalized Hessian of ψ at y. However, we caution the reader that it is unclear whether $\hat{\partial}^2\psi(y) = \partial^2\psi(y)$. Given any $y \in \mathbb{R}^m$, define

$$H := \tau\sigma^{-1}I_m + \sigma AUA^* \tag{3.13}$$

with $U \in \partial\Pi_K(\tilde{x} + \sigma(A^*y - c))$. Then, we know that $H \in \hat{\partial}^2\psi(y)$ and H is symmetric positive definite.

After these preparations, we are ready to present the following semismooth Newton method for solving the nonsmooth equation (3.12) and we can expect a fast local superlinear convergence.

Algorithm 3.2. SSN: A semismooth Newton method for solving (3.12) (SSN$(\tilde{x}, \tilde{y}, \sigma, \tau)$).

Given $\tau > 0, \sigma > 0$, choose parameters $\bar{\eta} \in (0, 1), \gamma \in (0, 1]$ and $\mu \in (0, 1/2), \delta \in (0, 1)$ and set $y^0 = \tilde{y}$. Iterate the following steps for $j = 0, 1, \ldots$.

Step 1. Choose $U_j \in \partial\Pi_K(\tilde{x} + \sigma(A^*y^j - c))$. Set $H_j := \tau\sigma^{-1}I_m + \sigma AU_jA^*$. Solve the linear system

$$H_jd = -\nabla\psi(y^j) \tag{3.14}$$

exactly or by a Krylov iterative method to find d^j such that $\|H_jd^j + \nabla\psi(y^j)\| \le \min(\bar{\eta}, \|\nabla\psi(y^j)\|^{1+\gamma})$.

Step 2. (Line search) Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\psi(y^j + \delta^m d^j) \le \psi(y^j) + \mu\delta^m\langle\nabla\psi(y^j), d^j\rangle.$$

Step 3. Set $y^{j+1} = y^j + \alpha_j d^j$.

The convergence results of the SSN algorithm are stated in the following theorem.

THEOREM 3.5. *Let $\{y^j\}$ be the infinite sequence generated by the* SSN *algorithm. It holds that $\{y^j\}$ converges to the unique optimal solution $\bar{y}$ of* (3.11) *and $\|y^{j+1} - \bar{y}\| = \mathcal{O}(\|y^j - \bar{y}\|^{1+\gamma})$.*

Proof. We know from [54, Proposition 3.3] that d^j is always a descent direction. Then, the strong convexity of ψ and [54, Theorem 3.4] imply that $\{y^j\}$ converges

to the unique optimal solution $\bar{y}$ of (3.11). By (3.13), we have that the symmetric positive definite matrix $H_j \in \hat{\partial}^2\psi(y^j)$ satisfies the property that $H_j \succeq \tau\sigma^{-1}I_m$ for all j. The desired results thus can be obtained by following the proof of [54, Theorem 3.5]. We omit the details here. ∎

3.3. Finite termination property of SNIPAL. In our numerical experience with SNIPAL, we observe that it nearly possesses a certain finite convergence property for solving (P) and (D) when σ_k and $1/\tau_k$ are sufficiently large. We note that most available theoretical results corresponding to the finite termination property of PPAs require each subproblem involved to be solved exactly, e.g., see [41, 42] and [29]. Hence, all these results cannot be directly adopted to support our numerical findings. In this section, we aim to investigate the finite termination property of SNIPAL by showing that it is possible to obtain a solution pair of (P) and (D) without requiring the exact solutions of each and every subproblem involved in the algorithm.

Our analysis is based on an interesting property called the "staircase property" associated with subdifferential mappings of convex closed polyhedral functions. Let

$$f(x) := c^T x + \delta_K(x) + \delta_{\{x|Ax=b\}}(x).$$

Clearly, f is a convex closed polyhedral function. From [18, section 6] and earlier work in [16, 29], we know that its subdifferential mapping enjoys the following staircase property, i.e., there exists $\delta > 0$ such that

$$w \in \partial f(x), \|w\| \le \delta \Rightarrow 0 \in \partial f(x). \tag{3.15}$$

Based on the staircase property of ∂f, we present the finite convergence property of SNIPAL in the following theorem.

THEOREM 3.6. *Suppose that Assumptions* 1 *and* 3 *hold and let* $\{(y^l, x^l)\}$ *be the infinite sequence generated by* SNIPAL *with the stopping criterion* (A′). *For any given* $k \ge 0$, *suppose that* $\bar{y}^{k+1}$ *is an exact solution to the following optimization problem:*

$$\bar{y}^{k+1} = \operatorname*{argmin}_{y\in\mathbb{R}^m} L_{\sigma_k}(y; x^k). \tag{3.16}$$

Then, the following results hold.

(a) *The point* $\bar{x}^{k+1} := \Pi_K\big(x^k - \sigma_k(c - A^*\bar{y}^{k+1})\big)$ *is the unique solution to the following proximal problem:*

$$\min\Big\{c^T x + \frac{1}{2\sigma_k}\|x - x^k\|^2 \mid Ax = b, x \in K\Big\}. \tag{3.17}$$

(b) *There exists a positive scalar* $\bar{\sigma}$ *independent of* k *such that for all* $\sigma_k \ge \bar{\sigma}$, $\bar{x}^{k+1}$ *also solves the problem* (P).

(c) *If* x^k *is a solution of* (P), *then* $\bar{y}^{k+1}$ *also solves* (D).

Proof. (a) Observe that the dual of (3.16) is exactly (3.17), and the KKT conditions associated with (3.16) and (3.17) are given as follows:

$$x = \Pi_K\big(x^k - \sigma_k(c - A^*y)\big), \quad Ax - b = 0. \tag{3.18}$$

Since $\bar{y}^{k+1}$ is a solution of the problem (3.16), it holds from the optimality condition associated with (3.16) that $A\Pi_K(x^k - \sigma_k(c - A^*\bar{y}^{k+1})) = b$. Thus, $(\bar{x}^{k+1}, \bar{y}^{k+1})$ satisfy

(3.18). Therefore, $\bar{x}^{k+1}$ solves (3.17). The uniqueness of $\bar{x}^{k+1}$ follows directly from the strong convexity of (3.17).

(b) By Theorem 3.3, we know that $x^l \to x^*$ as $l \to \infty$ for some $x^* \in \partial f^{-1}(0)$. Therefore, there exists a constant $M > 0$ (independent of k) such that

$$\|x^l - x^*\| \le M \quad \forall\, l \ge 0. \tag{3.19}$$

From the optimality of $\bar{x}^{k+1}$ and the definition of f, we have that

$$\frac{1}{\sigma_k}(x^k - \bar{x}^{k+1}) \in \partial f(\bar{x}^{k+1}).$$

It also holds from the nonexpansive property of the proximal mapping that $\|\bar{x}^{k+1} - x^*\| \le \|x^k - x^*\|$, which, together with (3.19), further implies that

$$\|\bar{x}^{k+1} - x^k\| \le 2\|x^k - x^*\| \le 2M.$$

Therefore, there exists $\bar{\sigma} > 0$ (independent of k) such that for all $\sigma_k \ge \bar{\sigma}$ and $k \ge 0$,

$$\frac{1}{\sigma_k}\|\bar{x}^{k+1} - x^k\| \le \frac{2M}{\bar{\sigma}} \le \delta,$$

where $\delta > 0$ is the constant given in (3.15). Thus, by using the staircase property (3.15), we know that

$$0 \in \partial f(\bar{x}^{k+1}).$$

That is, $\bar{x}^{k+1}$ solves the problem (P).

(c) Next, consider the case when x^k is a solution of (P). From the minimization property of x^k, it is clear that the unique solution of (3.17) must be $\bar{x}^{k+1} = x^k$. Thus, $x^k = \Pi_K\big(x^k - \sigma_k(c - A^*\bar{y}^{k+1})\big)$ and $Ax^k = b$. Note that it can be equivalently rewritten as

$$A^*\bar{y}^{k+1} - c \in \partial\delta_K(x^k), \quad Ax^k = b,$$

i.e., $(x^k, \bar{y}^{k+1})$ satisfy the KKT conditions for (P) and (D) in (3.1). Thus, $\bar{y}^{k+1}$ solves (D). ∎

Remark 4. We now remark on the significance of the above theorem. Essentially, it says that when σ_k is sufficiently large with $\sigma_k \ge \bar{\sigma}$, then $\bar{x}^{k+1}$ solves (P), and it holds that $\bar{y}^{k+2} = \operatorname{argmin} L_{\sigma_{k+1}}(y; \bar{x}^{k+1})$ solves (D).

From the fact that the SSN method used to solve (3.12) has the finite termination property [21, 46], we know that y^{k+1} computed in Step 1 of SNIPAL is in fact the exact solution of the subproblem $\min \psi_k(y)$ when the corresponding linear system is solved exactly. In addition, when σ_k is sufficiently large and τ_k is small enough, we have that

$$0 = \nabla L_{\sigma_k}(y^{k+1}; x^k) + \tau_k\sigma_k^{-1}(y^{k+1} - y^k) \approx \nabla L_{\sigma_k}(y^{k+1}; x^k),$$

and consequently, y^{k+1} can be regarded as a highly accurate solution to the problem $\min L_{\sigma_k}(y; x^k)$. In this sense, Theorem 3.6 explains the finite termination phenomenon in the practical performance of SNIPAL.

4. Solving the linear systems arising from the semismooth Newton method. Note that the most expensive operation in the SSN algorithm is the computation of the search direction $d \in \mathbb{R}^m$ through solving the linear system (3.14). To ensure the efficiency of SSN and consequently that of SNIPAL, in this section, we shall

discuss efficient approaches for solving (3.14) in the SSN Algorithm. Given $c, \tilde{x} \in \mathbb{R}^n$, $\tilde{y} \in \mathbb{R}^m$, the parameters $\tau, \sigma > 0$, and the current iterate of SSN $\hat{y} \in \mathbb{R}^m$, let

$$g := -\nabla\psi(\hat{y}) = R_p - \tau\sigma^{-1}(\hat{y} - \tilde{y}),$$

where $R_p = b - A\Pi_K(w(\hat{y}))$ with $w(\hat{y}) := \tilde{x} + \sigma(A^*\hat{y} - c)$. At each SSN iteration, we need to solve a linear system of the form

$$H\Delta y \;=\; g, \tag{4.1}$$

where $H = \tau\sigma^{-1}I_m + \sigma AUA^*$ with $U \in \partial\Pi_K(w(\hat{y}))$. Define the index set $\mathcal{J} = \{i \mid l_i < [w(\hat{y})]_i < u_i, i = 1, \ldots, n\}$ and $p = |\mathcal{J}|$, i.e., the cardinality of $\mathcal{J}$. In the implementation, we always construct the generalized Jacobian matrix $U \in \partial\Pi_K(w(\hat{y}))$ as a diagonal matrix in the following manner:

$$U = \mathrm{Diag}(u) \text{ with } u_i = \begin{cases} 1 & \text{if} \quad i \in \mathcal{J}, \\ 0 & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, n.$$

Without loss of generality, we can partition $A \equiv [A_{\mathcal{J}}, A_{\mathcal{N}}]$ with $A_{\mathcal{J}} \in \mathbb{R}^{m\times p}$, $A_{\mathcal{N}} \in \mathbb{R}^{m\times(n-p)}$, and hence

$$H = \sigma A_{\mathcal{J}}A_{\mathcal{J}}^* + \tau\sigma^{-1}I_m = \sigma(A_{\mathcal{J}}A_{\mathcal{J}}^* + \rho I_m), \tag{4.2}$$

where $\rho := \tau\sigma^{-2}$. To solve the linear system (4.1) efficiently, we need to consider various scenarios. In the discussion below, we use $\mathtt{nnzden}(M)$ to denote the density of the nonzero elements of a given matrix M.

(a) First, we consider the case where $p \geq m$ and the sparse Cholesky factorization of $A_{\mathcal{J}}A_{\mathcal{J}}^*$ can be computed at a moderate cost. In this case, the main cost of solving the linear system is in forming the matrix $A_{\mathcal{J}}A_{\mathcal{J}}^*$ at the cost of $O(m^2p\,\mathtt{nnzden}(A_{\mathcal{J}}))$ and computing the sparse Cholesky factorization of $A_{\mathcal{J}}A_{\mathcal{J}}^* + \rho I_m$.

Observe that the index set $\mathcal{J}$ generally changes from one SSN iteration to the next. However, when the SSN method is converging, the index set $\mathcal{J}$ may only change slightly from the current iteration to the next. In this case, one can update the inverse of H via a low-rank update by using the Sherman–Morrison–Woodbury formula [23, p. 65].

When it is expensive to compute and factorize H, one would naturally use a PCG method or the MINRES (minimim residual) method to solve (4.1). Note that in our implementation, we used the diagonal preconditioner for simplicity. For more elaborate numerical implementation in the future, one could explore more sophisticated preconditioners such as incomplete Cholesky factorizations or those proposed in [10, 45]. Observe that the condition number of H is given by $\kappa(H) = (\omega_{\max}^2 + \rho)/(\omega_{\min}^2 + \rho)$ if $p \geq m$, where $\omega_{\max}, \omega_{\min}$ are the largest and smallest singular value of $A_{\mathcal{J}}$, respectively. Note that when A is not explicitly given as a matrix, one can compute the matrix-vector product Hv as follows: $Hv = \sigma\rho v + \sigma A(e_{\mathcal{J}} \circ (A^*v))$, where $e_{\mathcal{J}} \in \mathbb{R}^n$ is a 0-1 vector whose nonzero entries are located at the index set $\mathcal{J}$, and "$\circ$" denotes the elementwise product.

(b) Next we consider the case where $p < m$. In this case, it is more economical to solve (4.1) by using the Sherman–Morrison–Woodbury formula to get

$$\Delta y = H^{-1}g = \tau^{-1}\sigma\big(I_m - P_{\mathcal{J}}\big)g, \tag{4.3}$$

where $P_{\mathcal{J}} = A_{\mathcal{J}}G^{-1}A_{\mathcal{J}}^*$, $G = \rho I_p + A_{\mathcal{J}}^*A_{\mathcal{J}} \in \mathbb{R}^{p\times p}$. Thus to compute Δy, one needs only to solve a smaller $p \times p$ linear system of equations $Gv = A_{\mathcal{J}}^*g$. Observe that when

ρ is close to zero, Δy is approximately the orthogonal projection of $\tau^{-1}\sigma g$ onto the null space of $A_{\mathcal{J}}^*$.

To solve (4.3), one can compute the sparse Cholesky factorization of the symmetric positive definite matrix $G \in \mathbb{R}^{p\times p}$ if the task can be done at a reasonable cost. In this case, the main cost involved in (4.3) is in computing $A_{\mathcal{J}}^* A_{\mathcal{J}}$ at the cost of $O(p^2 m\,\texttt{nnzden}(A_{\mathcal{J}}))$ operations and the sparse Cholesky factorization of $G = \rho I_p + A_{\mathcal{J}}^* A_{\mathcal{J}}$.

When it is too expensive to compute and factorize G, one can use a Krylov iterative method to solve the $p \times p$ linear system of equations:

$$Gv = (\rho I_p + A_{\mathcal{J}}^* A_{\mathcal{J}})v = A_{\mathcal{J}}^* g. \tag{4.4}$$

To estimate the convergence rate of the Krylov iterative method, it is important for us to analyze the conditioning of the above linear system, as is done in the next theorem.

THEOREM 4.1. *Let $B \in \mathbb{R}^{m\times p}$ with $p < m$. Consider linear system $Gv = B^* g$, where $G = B^* B + \rho I_p$ and $g \in \mathbb{R}^m$. Then the effective condition number for solving the system by the MINRES method with zero initial point is given by*

$$\kappa = \frac{\omega_{\max}^2 + \rho}{\omega_{\min}^2 + \rho},$$

where $\omega_{\max}$ is the largest singular value and $\omega_{\min} > 0$ is the smallest positive singular value of B, respectively.

Proof. Consider the following full SVD of B:

$$B = U\Sigma V^T = [U_1, U_2] \begin{bmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

where $\hat{\Sigma}$ is the diagonal matrix consisting of the positive singular values of B. Let $\mathbb{P}_k^0$ be the set of polynomials p_k with degree at most k and $p_k(0) = 1$. Then for $p_k \in \mathbb{P}_k^0$, we have that

$$\begin{aligned} p_k(G)B^* g &= V p_k(\Sigma^T\Sigma + \rho I)\Sigma^T U^T g = [V_1, V_2] \begin{bmatrix} p_k(\hat{\Sigma}^2 + \rho I)\hat{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T g \\ U_2^T g \end{bmatrix} \\ &= V_1 p_k(\hat{\Sigma}^2 + \rho I)\hat{\Sigma} U_1^T g. \end{aligned}$$

Since the kth iteration of the MINRES method computes an approximate solution x_k such that its residual $\xi = \bar{p}_k(G)B^* g$ satisfies the condition that

$$\|\xi\| = \|\bar{p}_k(G)B^* g\| \;=\; \min_{p_k \in \mathbb{P}_k^0} \|p_k(G)B^* g\| \;\le\; \|\hat{\Sigma} U_1^T g\| \min_{p_k \in \mathbb{P}_k^0} \|p_k(z)\|_{[\omega_{\min}^2 + \rho,\, \omega_{\max}^2 + \rho]},$$

we see that the convergence rate of the MINRES method is determined by the best approximation of the zero function by the polynomials in P_k^0 over the interval $[\omega_{\min}^2 + \rho, \omega_{\max}^2 + \rho]$. More specifically, by [44, Theorem 6.4], we have that $\min_{p_k \in \mathbb{P}_k^0} \|p_k(z)\|_{[\omega_{\min}^2 + \rho,\, \omega_{\max}^2 + \rho]} \le 2\kappa^{-k}$. Hence the convergence rate of the MINRES method is determined by κ. ∎

After (4.4) is solved via the MINRES method, one can compute the residual vector associated with system (4.3) without much difficulty. Indeed, let the computed solution of (4.3) be given as follows:

$$\widehat{\Delta y} = \tau^{-1}\sigma(g - A_{\mathcal{J}} v),$$

where $Gv = A_{\mathcal{J}}^* g - \xi$ with ξ being the residual vector obtained from the MINRES

iteration. Now the residual vector associated with (4.3) is given by

$$\begin{aligned}\eta &:= g - \mathcal{H}\widehat{\Delta y} = g - \tau^{-1}\sigma\mathcal{H}g + \tau^{-1}\sigma\mathcal{H}A_{\mathcal{J}}G^{-1}(A_{\mathcal{J}}^{*}g - \xi)\\ &= g - \tau^{-1}\sigma\mathcal{H}(g - P_{\mathcal{J}}g) - \tau^{-1}\sigma\mathcal{H}A_{\mathcal{J}}G^{-1}\xi\\ &= -\tau^{-1}\sigma\mathcal{H}A_{\mathcal{J}}G^{-1}\xi \;=\; -\rho^{-1}A_{\mathcal{J}}\xi,\end{aligned}$$

where the last equation follows directly from the fact that $\mathcal{H}A_{\mathcal{J}} = \sigma A_{\mathcal{J}}G$. Based on the computed η, one can check the termination condition for solving the linear system in (3.14).

Now, we are ready to bound the condition numbers of the Newton linear systems involved in SNIPAL. As can be observed from the above discussions, for both cases (a) and (b), the effective condition number of the linear system involved is upper bounded by

$$\kappa \le 1 + \frac{\omega_{\max}^2}{\rho},$$

where $\omega_{\max}$ is the largest singular value of $A_{\mathcal{J}}$ and $\rho = \tau\sigma^{-2}$. Since $A_{\mathcal{J}}$ is a submatrix of A, it holds that $\omega_{\max} \le \|A\|_2$. Hence, for any linear system involved in the kth iteration of SNIPAL, we can provide an upper bound for the condition number as follows:

$$\kappa \le 1 + \frac{\|A\|_2^2\sigma_k^2}{\tau_k}. \tag{4.5}$$

From our assumptions on SNIPAL, we note that $\sigma_k \le \sigma_\infty$ and $\tau_k \ge \tau_\infty > 0$ for all $k \ge 0$. Hence, for all the linear systems involved in SNIPAL, there exists a uniform upper bound for the corresponding condition number:

$$\kappa \le 1 + \frac{\|A\|_2^2\sigma_\infty^2}{\tau_\infty}.$$

As long as $\sigma_\infty < +\infty$, we have shown that all these linear systems have bounded condition numbers. This differs significantly from the setting in interior-point based algorithms where the condition numbers of the corresponding normal equations are asymptotically unbounded as the barrier parameter tends to 0. The competitive advantage of SNIPAL can be partially explained from the above observation. Meanwhile, in the kth iteration of SINPAL, to get a smaller upper bound based on (4.5), one should choose a small σ_k but large τ_k. However, the convergence rate of SNIPAL developed in Theorem 2.5 requires the opposite choice, i.e., a large σ_k and τ_k should be moderate. The preceding discussion thus reveals the trade-off between the convergence rate of the ALM and the condition numbers of the Newton linear systems. Clearly, in the implementation of SNIPAL, the parameters $\{\sigma_k\}$ and $\{\tau_k\}$ should be chosen to balance the progress of the outer and inner algorithms, i.e., the ALM and the semismooth Newton method.

5. Warm-start algorithm for SNIPAL. As is mentioned in the introduction, to achieve high performance, it is desirable to use a simple first-order algorithm to warm-start SNIPAL so that its local linear convergence behavior can be observed earlier. For this purpose, we present an ADMM algorithm for solving (D). We note that a similar strategy has also been employed for solving large scale semidefinite programming and quadratic semidefinite programming problems [51, 28].

We begin by rewriting (D) into the following equivalent form:

$$\min\left\{\delta_K^*(-z) - b^T y \mid z + A^* y = c\right\}. \tag{5.1}$$

Given $\sigma > 0$, the augmented Lagrangian function associated with (5.1) can be written as

$$\mathbf{L}_\sigma(z, y; x) = \delta_K^*(-z) - b^T y + \langle x,\, z + A^* y - c\rangle + \frac{\sigma}{2}\|z + A^* y - c\|^2$$

for all $(x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$. The template of the classical ADMM for solving (5.1) is given as follows.

Algorithm 5.1. ADMM. An ADMM method for solving (5.1).

Given $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m$ and $\gamma > 0$, perform the following steps for $k = 1, \ldots,$

Step 1. Compute

$$\begin{aligned} z^{k+1} &= \operatorname{argmin} \mathbf{L}_\sigma(z, y^k; x^k) \\ &= \frac{1}{\sigma}\left(\Pi_K(x^k + \sigma(A^* y^k - c)) - (x^k + \sigma(A^* y^k - c))\right). \end{aligned} \tag{5.2}$$

Step 2. Compute

$$y^{k+1} = \operatorname{argmin} \mathbf{L}_\sigma(z^{k+1}, y; x^k) \;=\; (AA^*)^{-1}\Big(b/\sigma - A(x^k/\sigma + z^{k+1} - c)\Big). \tag{5.3}$$

Step 3. Compute $x^{k+1} = x^k + \gamma\sigma(z^{k+1} + A^* y^{k+1} - c)$.

The convergence of the above classical ADMM for solving the two-block optimization problem (5.1) with the steplength $\gamma \in (0, (1+\sqrt{5})/2)$ can be readily obtained from the vast literature on ADMM. Here, we adopt a newly developed result from [12] stating that the above ADMM is in fact an inexact proximal ALM. This new interpretation allows us to choose the steplength γ in the larger interval $(0, 2)$, which usually leads a better numerical performance when γ is chosen to be 1.9 instead of 1.618. We summarize the convergence results in the following theorem. A detailed proof can be found in [12].

THEOREM 5.1. *Suppose that Assumption* 1 *holds and* $\gamma \in (0, 2)$. *Let* $\{(x^k, y^k, z^k)\}$ *be the sequence generated by the ADMM algorithm. Then,* $\{x^k\}$ *converges to an optimal solution of (P) and* $\{(y^k, z^k)\}$ *converges to an optimal solution of* (5.1)*, respectively.*

Remark 5. In the above algorithm, one can also handle (5.3) by adding an appropriate proximal term or by using an iterative method (with appropriate preconditioning techniques) to solve the corresponding linear system. The convergence of the resulting proximal or inexact ADMM with steplength $\gamma \in (0, 2)$ has also been discussed in [12]. For simplicity, we only discussed the exact version here.

6. Numerical experiments. In this section, we evaluate the performance of SNIPAL against the powerful commercial solver Gurobi (version 8.0.1) on various LP data sets. Our goal is to compare the performance of our algorithm against the barrier method implemented in Gurobi in terms of its speed and ability to solve the tested instances to the relatively high accuracy of 10^{-6} or 10^{-8} in the relative KKT residual. That is, for a given computed solution (x, y, z), we stop the algorithm when

$$\eta = \max\left\{\frac{\|b - Ax\|}{1 + \|b\|},\, \frac{\|A^T y + z - c\|}{1 + \|c\|},\, \frac{\|x - \Pi_K(x - z)\|}{1 + \|x\| + \|z\|}\right\} \;\le\; \texttt{Tol}, \tag{6.1}$$

where `Tol` is a given accuracy tolerance. We should note that it is possible to solve an LP by using the primal or dual simplex methods in Gurobi, and those methods

could sometimes be more efficient than the barrier method in solving large scale LPs. However, as our SNIPAL algorithm is akin to a barrier method, in that each of its semismooth Newton iteration also requires the solution of a linear system having the form of normal equations just as in the case of the barrier method, we thus restrict the comparison of our algorithm only to the barrier method in Gurobi. To purely use the barrier method in Gurobi, we also turn off its crossover capability from the barrier method to simplex methods. We should note that sometimes the presolve phrase in Gurobi is too time consuming and does not lead to any reduction in the problem size. In that case, we turn off the presolve phase in Gurobi to get the actual performance of its barrier method.

All the numerical experiments in this paper were run in MATLAB on a Dell laptop with Intel Core i7-6820HQ CPU @2.70GHz and 16GB of RAM. As Gurobi is extremely powerful in exploiting multithread computing, we set the number of threads allowed for Gurobi to be two so that its overall CPU utilization rate is roughly the same as that observed for running SNIPAL in MATLAB when setting the maximum number of computational threads to be two.

In our experiments, unless otherwise stated, we adopt the following numerical strategies for solving the Newton linear system (4.1) in each iteration. (a) If $p \geq m/2$, solve (4.1) as follows. For the case when $m \leq 30000$ and the density of the nonzero elements of H is less than 30%, solve (4.1) via sparse Cholesky factorization; otherwise solve (4.1) by the MINRES iterative solver with diagonal preconditioning. (b) If $p < m/2$, solve (4.1) via the Sherman–Morrison–Woodbury update (4.3) as follows. For the case when $p \leq 30000$ and the density of the nonzero elements of G is less than 30%, solve (4.4) via sparse Cholesky factorization; otherwise solve (4.4) by the MINRES iterative solver with diagonal preconditioning.

6.1. Randomly generated sparse LP in [32]. Here we test large synthetic LP problems generated as in [32]. In particular, the matrix A is generated as follows:

```
rng('default');  A = sprand(m,n,d);  A = 100*(A-0.5*spones(A));
```

In this case, we turn off the presolve phase in Gurobi as this phase is too time consuming for these randomly generated problems and it also does not lead to any reduction in the problem sizes. As we can observe from Table 1, SNIPAL is able to outperform Gurobi by a factor of about 1.5–2.3 in computational time in most cases.

Note that for the column "iter (itssn)" in Table 1, we report the number of SNIPAL iterations and the total number of semismooth Newton linear systems solved in Algorithm 2. For the columns "time (RAM)" and "Gurobi time (RAM)," we report the wall-clock time and the memory consumed by SNIPAL and Gurobi, respectively.

6.2. Transportation problem. In this problem, s suppliers of $a_1, \ldots, a_s$ units of certain goods must be transported to meet the demands $b_1, \ldots, b_t$ of t customers. Let the cost of transporting one unit of goods from supplier i to customer j be c_{ij}. Then, the objective is to find a transportation plan denoted by x_{ij} to solve the following LP:

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{s} \sum_{j=1}^{t} c_{ij} x_{ij} \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{t} x_{ij} = a_i, \quad i \in [s], \\ & \textstyle\sum_{i=1}^{s} x_{ij} = b_j, \quad j \in [t], \\ & x_{ij} \geq 0 \ \forall\, i \in [s],\ j \in [t]. \end{aligned}$$

TABLE 1
Numerical results for random sparse LPs with $\texttt{Tol} = 10^{-8}$.

m	n	d	SNIPAL iter (itssn)	SNIPAL time (s) (RAM)	Gurobi barrier iter	Gurobi time (s) (RAM)
2e3	1e5	0.025	4 (18)	3.5 (0.8GB)	7	5.2 (1.0GB)
		0.050	4 (18)	6.8 (0.8GB)	7	10.5 (1.3GB)
5e3	1e5	0.025	4 (15)	15.0 (1.5GB)	7	22.4 (1.7GB)
		0.050	4 (15)	31.1 (1.8GB)	7	46.1 (2.6GB)
10e3	1e5	0.025	5 (24)	50.6 (3.2GB)	8	96.6 (3.2GB)
		0.050	5 (24)	101.5 (5.3GB)	8	181.6 (6.0GB)
1e3	1e6	0.025	5 (30)	10.3 (2.0GB)	7	22.2 (4.1GB)
		0.050	5 (29)	18.9 (3.2GB)	7	40.0 (6.0GB)
2e3	1e6	0.025	6 (32)	27.2 (3.2GB)	7	52.2 (6.1GB)
		0.050	5 (28)	53.1 (5.2GB)	6	92.7 (9.6GB)
5e3	1e6	0.025	5 (26)	91.8 (4.5GB)	7	184.1 (10.0GB)
10e3	1e6	0.010	7 (40)	84.6 (4.3GB)	7	194.4 (8.5GB)

In the above problem, we assume that $\sum_{i=1}^{s} a_i = \sum_{j=1}^{t} b_j$. Note that this assumption is needed for the LP to be feasible. We can write the transportation LP compactly as follows:

$$\min\Big\{\langle C,\, X\rangle \mid \mathcal{A}(X) = [a; b],\ X \geq 0\Big\}, \tag{6.2}$$

where

$$\mathcal{A}(X) = \begin{bmatrix} \hat{e}^T \otimes I_s \\ I_t \otimes e^T \end{bmatrix} \mathrm{vec}(X),$$

$e \in \mathbb{R}^s$, and $\hat{e} \in \mathbb{R}^t$ are vectors of all ones, and $\mathrm{vec}(X)$ is the st-dimensional column vector obtained from X by concatenating its columns sequentially.

In Table 2, we report the results for some randomly generated transportation instances. For each pair of given s, t, we generate a random transportation instance as follows:

```
rng('default'); M=abs(rand(s,t)); a=sum(M,2); b=sum(M,1)';
C=ceil(100*rand(s,t));
```

Note that we turn off the presolve phase in Gurobi as this phase is too time consuming (about 20–30% of the total time) and there is no benefit in cutting down the computation time per iteration.

We can observe that for this class of problems, SNIPAL is able to outperform the highly powerful barrier method in Gurobi by a factor of about 1–3 in terms of computation times. Moreover, our solver SNIPAL consumed less peak memory than Gurobi. For the largest instance where the primal LP has 12,000 linear constraints and 27 millions variables, our solver is at least five times faster than the barrier method in Guorbi, and it only needs 5.4GB of RAM whereas Gurobi required 12.8GB.

6.3. Generalized transportation problem. The generalized transportation problem was introduced by Fergusan and Dantzig [20] in their study of an aircraft routing problem. Eisemann and Lourie [22] applied it to the machine loading problem. In that problem, there are m types of machines that can produce n types of products such that machine i would take h_{ij} hours at the cost of c_{ij} to produce one unit of product j. It is assumed that machine i is available for at most a_i hours, and the demand for product j is b_j. The problem is to determine x_{ij}, the amount of product

TABLE 2
Numerical results for transportation LPs with $\texttt{Tol} = 10^{-8}$.

s	t	SNIPAL iter (itssn)	SNIPAL time (s) (RAM)	Gurobi barrier iter	Gurobi time (s) (RAM)
2000	3000	5 (17)	18.3 (1.8GB)	8	20.7 (4.8GB)
2000	4000	5 (18)	22.0 (2.1GB)	8	32.6 (6.5GB)
2000	6000	5 (18)	34.2 (3.4GB)	8	59.5 (8.9GB)
3000	4500	5 (17)	40.4 (3.5GB)	8	61.6 (9.2GB)
3000	6000	5 (18)	53.4 (4.0GB)	8	93.9 (10.3GB)
3000	9000	5 (20)	65.1 (5.4GB)	7	191.1 (12.8GB)

TABLE 3
Numerical results for generalized transportation LPs with $\texttt{Tol} = 10^{-8}$.

s	t	SNIPAL iter (itssn)	SNIPAL time (s) (RAM)	Gurobi barrier iter	Gurobi time (s) (RAM)
2000	3000	5 (19)	22.4 (1.6GB)	7	19.4 (2.9GB)
2000	4000	5 (18)	27.1 (2.5GB)	8	32.7 (5.5GB)
2000	6000	5 (18)	40.6 (3.6GB)	8	61.0 (8.0GB)
3000	4500	5 (18)	48.4 (3.4GB)	8	59.7 (6.0GB)
3000	6000	5 (18)	63.5 (4.0GB)	8	90.0 (12.9GB)
3000	9000	5 (19)	85.3 (5.8GB)	7	258.0 (13.1GB)

j to be produced on machine i during the planning period so that the total cost is minimized, namely,

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{s}\sum_{j=1}^{t} c_{ij}x_{ij} \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{t} h_{ij}x_{ij} = a_i, \quad i \in [s], \\ & \textstyle\sum_{i=1}^{s} x_{ij} = b_j, \quad j \in [t], \\ & x_{ij} \geq 0 \ \forall\, i \in [s],\ j \in [t]. \end{aligned}$$

In addition to assuming, similar to the transportation problem in the previous subsection, that $\sum_{i=1}^{s} a_i = \sum_{j=1}^{t} b_j$, we also apply the normalization $\sum_{i=1}^{s}\sum_{j=1}^{t} h_{ij} = st$.

Table 3 presents the results for randomly generated generalized transportation LPs where a, b, c are generated as in the last subsection. The weight matrix $H = (h_{ij})$ is generated by setting `H = rand(s,t); H = (s*t/sum(sum(H)))*H`. We can observe that SNIPAL can be up to 3 times faster than the barrier method in Gurobi when the problems are large.

6.4. Covering and packing LPs. Given a nonnegative matrix $A \in \mathbb{R}^{m\times n}$ and cost vector $c \in \mathbb{R}^n_+$, the covering and packing LPs [5, Section 10.1] are defined by

$$\text{(Covering)}\min\Big\{\langle c,\, x\rangle \mid Ax \geq e, x \geq 0\Big\},$$

$$\text{(Packing)}\min\Big\{\langle -c,\, x\rangle \mid Ax \leq e, x \geq 0\Big\}.$$

It is easy to see that by adding a slack variable, the above problems can be converted into the standard form expressed in (P).

In our numerical experiments in Table 4, we generate A and c randomly as follows:

```
rng('default'); c = rand(n,1); A = sprand(m,n,den); A = round(A);
```

Table 4 presents the numerical performance of SNIPAL versus Gurobi on some randomly generated large scale covering and packing LPs. As we can observe, SNIPAL is

TABLE 4
Numerical results for covering and packing LPs with Tol $= 10^{-8}$.

Type	m	n	den	SNIPAL iter (itssn)	SNIPAL time (s)	Gurobi barrier iter	Gurobi time (s)
C	1e3	5e5	0.2	22 (148)	49.8	14	62.1
C	2e3	5e5	0.1	25 (151)	103.3	16	90.6
C	2e3	1e6	0.05	24 (160)	90.4	17	102.0
C	3e3	5e6	0.02	24 (148)	190.5	22	560.5
P	1e3	5e5	0.2	28 (160)	49.3	12	53.3
P	2e3	5e5	0.1	29 (160)	97.0	12	68.2
P	2e3	1e6	0.05	30 (173)	75.1	15	91.4
P	3e3	5e6	0.02	26 (228)	259.8	20	500.2

competitive against the barrier method in Gurobi for solving these large scale LPs, and the former can be up to 2.9 times faster than the barrier method in Gurobi.

6.5. LPs arising from correlation clustering. A correlation clustering problem [1] is defined over an undirected graph $G = (V, E)$ with p nodes and edge weights $c_e \in \mathbb{R}$ (for each $e \in E$) that is interpreted as a confidence measure of the similarity or dissimilarity of the edge's end nodes. In general, for $e = (u, v) \in E$, c_e is given a negative value if u, v are dissimilar and a positive value if u, v are similar. For the goal of finding a clustering that minimizes the disagreements, the problem can be formulated as an integer programming problem as follows. Suppose that we are given a clustering $\mathbb{S} = \{S_1, \ldots, S_N\}$, where each $S_t \subset V$, $t = 1, \ldots, N$, denotes a cluster. For each edge $e = (u, v) \in E$, set $y_e = 0$ if $u, v \in S_t$ for some t, and set $y_e = 1$ otherwise. Observe that $1 - y_e$ is 1 if u, v are in the same cluster and 0 if u, v are in different clusters. Now define the constants

$$m_e = |\min\{0, c_e\}|, \quad p_e = \max\{0, c_e\}.$$

Then the cost of disagreements for the clustering $\mathbb{S}$ is given by $\sum_{e\in E} m_e(1 - y_e) + \sum_{e\in E} p_e y_e$.

A version of the correlation clustering problem is to find a valid assignment (i.e., it satisfies the triangle inequalities) of y_e for all $e \in E$ to minimize the disagreements' cost. We consider the relaxation of this integer program to get the following LP [11]:

$$\begin{aligned} \min \quad & \textstyle\sum_{(i,j)\in E} m_{ij}(1 - y_{ij}) + \sum_{(i,j)\in E} p_{ij} y_{ij} \\ \text{s.t.} \quad & -y_{ij} \le 0, \;\; y_{ij} \le 1 \quad \forall\, (i,j) \in E, \\ & -y_{ij} - y_{jk} + y_{ik} \le 0 \quad \forall\, 1 \le i < j < k \le n, \text{ such that } (i,j), (j,k), (i,k) \in E. \end{aligned}$$

In the above formulation, we assumed that the edge set E is a subset of $\{(i, j) \mid 1 \le i < j \le p\}$. Let M be the number of all possible triangles in E. Define $\mathcal{T} : \mathbb{R}^{|E|} \to \mathbb{R}^M$ to be the linear map that maps y to all the M terms $-y_{ij} - y_{jk} + y_{ik}$ in the triangle inequalities. We can express the above LP in the dual form as follows,

$$\langle m,\, \mathbf{1}\rangle - \max\left\{ \langle m - p,\, y\rangle \mid \begin{bmatrix} -I \\ I \\ \mathcal{T} \end{bmatrix} y \;\le\; \begin{bmatrix} 0 \\ \mathbf{1} \\ 0 \end{bmatrix} \right\},$$

and the corresponding primal LP is given by

$$\langle m,\, \mathbf{1}\rangle - \min\left\{ \langle [0; \mathbf{1}; 0],\, x\rangle \mid [-I,\, I,\, \mathcal{T}^*]x = m - p,\; x \in \mathbb{R}^{2|E|+M}_+ \right\}. \tag{6.3}$$

TABLE 5
Numerical results for correlation clustering LPs with $\texttt{Tol} = 10^{-8}$.

Data	p	$\lvert E\rvert$	$2\lvert E\rvert + M$	SNIPAL iter (itssn | itminres)	SNIPAL time (s)	Gurobi barrier iter	Gurobi time (s)
planted(5)	200	19900	1353200	5 (70 | 110.0)	38.1	37	690.9
planted(10)	200	19900	1353200	6 (91 | 146.5)	36.8	49	1146.6
planted(5)	300	44850	4544800	5 (86 | 109.2)	170.2	37	8350.7
planted(10)	300	44850	4544800	7 (127 | 186.3)	158.0	82	18615.8
stocks	200	19900	1353200	5 (57 | 147.7)	57.8	53	1009.2
stocks	300	44850	4544800	5 (75 | 191.1)	276.9	60	13797.0

Observe that the primal LP has $|E|$ equality constraints and a large number of $2|E|+M$ variables.

In Table 5, we evaluate the performance of our algorithm on correlation clustering LPs on data that were used in [49]. Note that for this class of LPs, we solve the linear system (4.1) by the MINRES iterative solver with diagonal preconditioning. One can observe that for the LP problem (6.3), our solver SNIPAL is much more efficient than the barrier method in Gurobi, and the former can be up to 117 times faster for the largest problem. The main reason why SNIPAL is able to outperform the barrier method in Gurobi lies in the fact that the former is able to make use of an iterative solver to solve the moderately well conditioned linear system involving the matrix (4.2) rather efficiently in each semismooth Newton iteration, whereas for the latter, it has to rely on sparse Cholesky factorization to solve the associated normal equation and for this class of problems, computing the sparse Cholesky factorization is expensive. Under the column "itminres" in Table 5, we report the average number of MINRES iterations needed to solve a single linear system of the form in (4.2). As one can observe, the average number of MINRES iterations is small compared to the dimension of the linear system for all the tested instances.

6.6. LPs from MIPLIB2010. In this subsection, we evaluate the potential of SNIPAL as a tool for solving general LPs with the characteristic that the number of linear constraints is much smaller than the dimension of the variables. For this purpose, we consider the root-node LP relaxations of some mixed-integer programming problems in the library MIPLIB2010 [33].

Table 6 reports the performance of SNIPAL against the barrier method in Gurobi for solving the LPs from the two sources mentioned in the last paragraph to the accuracy level of 10^{-6}. Note that we first use Gurobi's presolve function to preprocess the LPs. Then the preprocessed instances are used for comparison with Gurobi's presolve capability turned off. As one can observe, the barrier method in Gurobi performed much better than SNIPAL, with the former typically requiring less than 50 iterations to solve the LPs while the latter typically needs hundreds of semismooth Newton iterations except for a few problems such as `datt256, neos-xxxx`, etc. Overall, the barrier method in Gurobi can be 10–50 times faster than SNIPAL on many of the tested instances, with the exception of `ns2137859`.

The large number of semismooth Newton iterations needed by SNIPAL to solve the LPs can be attributed to the fact that for most of the LP instances tested here, the local superlinear convergent property of the semismooth Newton method in solving the subproblems of the SPALM generally does not kick-in before a large number of initial iterations has been taken. From this limited set of tested LPs, we may conclude that substantial numerical work must be done to improve the practical performance of SNIPAL before it is competitive enough to solve general large scale sparse LPs.

TABLE 6
Numerical results for some LPs from MIPLIB2010 *with* `Tol` $= 10^{-6}$.

Problem	m	n	it (itssn)	Time	Gurobi barrier iter	Gurobe time
app1-2	26850	107132	33 (878)	54.33	16	0.73
bab3	22449	411334	40 (789)	139.43	37	6.41
bley-xl1	746	7361	7 (114)	1.06	21	0.21
circ10-3	2700	46130	7 (17)	4.98	10	0.80
co-100	1293	22823	39 (634)	4.09	23	0.64
core2536-691	1895	12991	11 (325)	15.35	25	0.83
core4872-1529	3982	18965	16 (285)	34.37	22	1.52
datt256	9809	193639	3 (37)	31.24	5	1.69
dc1l	1071	34931	16 (209)	8.86	39	1.06
ds-big	1039	173026	27 (623)	74.91	25	5.37
eilA101.2	100	65832	14 (88)	7.71	21	0.96
ivu06-big	1177	2197774	19 (236)	87.28	27	34.97
ivu52	2116	135634	31 (559)	47.19	23	3.22
lectsched-1-obj	9246	34592	28 (331)	2.46	12	0.32
lectsched-1	6731	27042	7 (15)	0.26	5	0.15
lectsched-4-obj	2592	9716	22 (94)	0.76	7	0.06
leo2	539	11456	24 (106)	0.68	22	0.16
mspp16	4065	532749	26 (54)	68.67	14	59.88
n3div36	4450	25052	20 (75)	0.72	27	0.25
n3seq24	5950	125746	14 (71)	20.25	15	6.42
n15-3	29234	153400	22 (475)	63.86	30	4.30
neos13	1826	22930	30 (154)	9.83	22	0.23
neos-476283	9227	20643	22 (495)	174.60	14	10.58
neos-506428	40806	200653	4 (8)	7.86	16	0.96
neos-631710	3072	169825	4 (9)	13.90	5	0.54
neos-885524	60	21317	4 (8)	0.84	12	0.11
neos-932816	2568	8932	7 (17)	2.88	10	0.14
neos-941313	12919	129180	6 (17)	5.71	10	0.42
neos-1429212	8773	42620	37 (541)	118.34	28	6.91
netdiversion	99482	208447	33 (324)	173.07	15	5.38
ns1111636	12992	85327	4 (38)	7.85	16	0.79
ns1116954	11928	141529	2 (6)	125.54	11	23.86
ns1688926	16489	41170	26 (160)	150.80	88	12.68
ns1904248	38184	222489	3 (6)	6.00	6	0.96
ns2118727	7017	15853	30 (1079)	20.63	24	0.41
ns2124243	19663	53716	22 (122)	13.08	14	0.36
ns2137859	16357	49795	11 (22)	6.20	50	61.53
opm2-z12-s7	10328	145436	13 (43)	19.13	17	16.19
opm2-z12-s14	10323	145261	12 (36)	19.46	16	15.39
pb-simp-nonunif	11706	146052	2 (4)	2.08	10	0.68
rail507	449	23161	14 (240)	1.95	23	0.25
rocII-7-11	5534	25590	20 (73)	2.26	17	0.35
rocII-9-11	8176	37159	22 (106)	3.85	17	0.51
rvb-sub	217	33200	24 (157)	1.67	11	0.56
shipsched	5165	22806	16 (35)	0.73	10	0.24
sp97ar	1627	15686	26 (264)	2.63	26	0.33
sp98ic	806	11697	27 (155)	1.63	25	0.25
stp3d	95279	205516	71 (2892)	228.56	22	11.75
sts729	729	89910	2 (4)	0.72	3	0.26
t1717	551	16428	22 (141)	1.59	13	0.22
tanglegram1	32705	130562	2 (4)	0.90	5	0.30
van	7360	36736	4 (8)	7.39	15	3.63
vpphard	9621	22841	3 (6)	2.90	9	0.90
vpphard2	13085	28311	4 (6)	2.77	7	0.64
wnq-n100-mw99-14	594	10594	24 (119)	0.73	15	0.22

7. Conclusion. In this paper, we proposed a method called SNIPAL targeted at solving large scale LP problems where the dimension n of the decision variables is much larger than the number m of equality constraints. SNIPAL is an inexact proximal augmented Lagrangian method where the inner subproblems are solved via an efficient semismooth Newton method. By connecting the inexact proximal augmented Lagrangian method with the preconditioned proximal point algorithm, we are able to show the global and local asymptotic superlinear convergence of SNIPAL. Our analysis also reveals that SNIPAL can enjoy a certain finite termination property. To achieve high performance, we further study various efficient approaches for solving the large linear systems in the semismooth Newton method. Our findings indicate that the linear systems involved in SNIPAL can have uniformly bounded condition numbers with respect to the parameter sequences $\{\sigma_k\}$ and $\{\tau_k\}$ when they are chosen such that $\sup\{\sigma_k\} < \infty$ and $\inf\{\tau_k\} > 0$. This is in contrast to those involved in an interior point algorithm which has unbounded condition numbers with respect to the barrier parameter sequence that must be driven to zero. Building upon all the aforementioned desirable properties, our SNIPAL algorithm has demonstrated a clear computational advantage in solving some classes of large-scale LP problems in the numerical experiments when tested against the barrier method in the powerful commercial LP solver Gurobi. However, when tested on some large sparse LPs available in the public domain, SNIPAL is not yet competitive against the barrier method in Gurobi on most of the test instances. Thus much work remains to be done to improve the practical efficiency of SNIPAL and we leave it as a future research project.

Appendix. Here we show that the dual of (3.2) with $\tau = 0$ is given by (3.17). Consider the augmented Lagrangian function

$$\begin{aligned}\inf_y L_{\sigma_k}(y; x^k) &= \inf_y \max_x \left\{ l(y; x) - \frac{1}{2\sigma}\|x - x^k\|^2 \right\} \\ &= \max_x \left\{ -\frac{1}{2\sigma}\|x - x^k\|^2 + \inf_y l(y; x) \right\} \\ &= \max_x \Big\{ -\delta_K(x) - \langle c,\, x\rangle - \frac{1}{2\sigma}\|x - x^k\|^2 \mid Ax = b \Big\},\end{aligned}$$

where $l(y; x) = -b^T y - \langle x,\, c - A^* y\rangle - \delta_K(x)$ for any $(y, x) \in \mathbb{R}^m \times \mathbb{R}^n$. The interchange of $\inf_y$ and $\max_x$ follows from the growth properties in x of the "minimaximand" in question [40, Theorem 37.3]. See also the proof of [42, Proposition 6].

REFERENCES

[1] N. BANSAL, A. BLUM, AND S. CHAWLA, *Correlation clustering*, Proceedings of the IEEE Symposium on Foundations of Computer Science, 2002.

[2] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.

[3] L. BERGAMASCHI, J. GONDZIO, AND G. ZILLI, *Preconditioning indefinite systems in interior point methods for optimization*, Comput. Optim. Appl., 28 (2004), pp. 149–171.

[4] L. BERGAMASCHI, J. GONDZIO, A. MARTÍNEZ, J. W. PEARSON, AND S. POUGKAKIOTIS, *A New Preconditioning Approach for an Interior Point Proximal Method of Multipliers for Linear and Convex Quadratic Programming*, preprint, https://arxiv.org/abs/1912.10064, 2019.

[5] D. BERTSIMAS AND J. N. TSITSIKLIS, *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.

[6] J. F. BONNANS, J. CH. GILBERT, C. LEMARÉCHAL, AND C. A. SAGASTIZÁBAL, *A family of variable metric proximal methods*, Math. Program., 68 (1995), pp. 15–47.

[7] J. V. Burke and M. Qian, *A variable metric proximal point algorithm for monotone operators*, SIAM J. Control Optim., 37 (1999), pp. 353–375.
[8] J. V. Burke and M. Qian, *On the local super-linear convergence of a matrix secant implementation of the variable metric proximal point algorithm for monotone operators*, in Reformulation—Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, M. Fukushima and L. Qi, eds., Kluwer Academic Publishers, Norwell, MA, 1999, pp. 317–334.
[9] J. V. Burke and M. Qian, *On the superlinear convergence of the variable metric proximal point algorithm using Broyden and BFGS matrix secant updating*, Math. Program., 88 (2000), pp. 157–181.
[10] J. S. Chai and K.-C. Toh, *Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming*, Comput. Optim. Appl., 36 (2007), pp. 221–247.
[11] M. Charikar, V. Guruswami, and A. Wirth, *Clustering with qualitative information*, J. Comput. System Sci., 71 (2005), pp. 360–383.
[12] L. Chen, X. D. Li, D. F. Sun, and K.-C. Toh, *On the equivalence of inexact proximal ALM and ADMM for a class of convex composite programming*, Math. Program., 2019, https://doi.org/10.1007/s10107-019-01423-x.
[13] X. Chen and M. Fukushima, *Proximal quasi-Newton methods for nondifferentiable convex optimization*, Math. Program., 85 (1999), pp. 313–334.
[14] F. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.
[15] Y. Cui, K. Morikuni, T. Tsuchiya, and K. Hayami, *Implementation of interior-point methods for LP based on Krylov subspace iterative solvers with inner-iteration preconditioning*, Comput. Optim. Appl., 74 (2019), pp. 143–176.
[16] R. Durier, *On locally polyhedral convex functions*, in Trends in Mathematical Optimization (Irsee, 1986), Internat. Schriftenreihe Numer. Math. 84, Birkhäuser, Basel, 1988, pp. 55–66.
[17] J. Eckstein, *Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming*, Math. Oper. Res., 18 (1993), pp. 202–226.
[18] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., 55 (1992), pp. 293–318.
[19] Yu. G. Evtushenko, A. I. Golikov, and N. Mollaverdy, *Augmented Lagrangian method for large-scale linear programming problems*, Optim. Methods Softw., 20 (2005), pp. 515–524.
[20] A. R. Fergusan and G. B. Dantzig, *The allocation of aircrafts to routes—An example of linear programming under uncertain demand*, Manag. Sci., 3 (1956).
[21] A. Fischer and C. Kanzow, *On finite termination of an iterative method for linear complementarity problems*, Math. Program., 74 (1996), pp. 279–292.
[22] K. Eisemann and J. R. Lourie, *The Machine Loading Problem*, IBM 704 program, BML-1, IBM Application Library, New York, 1959.
[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., The Johns Hopkins University Press, Baltimore, MD, 2013.
[24] G. Al-Jeiroudi, J. Gondzio and J. A. J. Hall, *Preconditioning indefinite systems in interior point methods for large scale linear optimization*, Optim. Methods Softw., 23 (2008), pp. 345–363.
[25] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, *Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data*, Appl. Math. Optim., 11 (1984), pp. 43–56.
[26] C. Kanzow, H. Qi, and L. Qi, *On the minimum norm solution of linear programs*, J. Optim. Theory Appl., 116 (2003), pp. 333–345.
[27] D. Klatte and B. Kummer, *Nonsmooth Equations in Optimization, Regularity, Calculus, Methods and Applications*, Kluwer Academic Publishers, Belmont, MA, 2002.
[28] X. D. Li, D. F. Sun, and K.-C. Toh, *QSDPNAL: A two-phase augmented Lagrangian method for convex quadratic semidefinite programming*, Math. Program. Comput., 10 (2018), pp. 703–743.
[29] F. J. Luque, *Asymptotic convergence analysis of the proximal point algorithm*, SIAM J. Control Optim., 22 (1984), pp. 277–293.
[30] R. De Leone and O. L. Mangasarian, *Serial and parallel solution of large scale linear programs by augmented Lagrangian successive overrelaxation*, in Optimization, Parallel Processing and Applications, A. Kurzhanski, K. Neumann, and D. Pallaschke, eds., pp. 103–124, Springer, New York, 1988.
[31] O. L. Mangasarian and R. R. Meyer, *Nonlinear perturbation of linear programs*, SIAM J. Control Optim., 17 (1979), pp. 745–752.

[32] O. L. Mangasarian, *A Newton method for linear programming*, J. Optim. Theory Appl., 121 (2004), pp. 1–18.
[33] *MIPLIB—the Mixed Integer Programming LIBrary*, http://miplib2010.zib.de/.
[34] A. R. L. Oliveira and D. C. Sorensen, *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra Appl., 394 (2005), pp. 1–24.
[35] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.
[36] L. A. Parente, P. A. Lotito, and M. V. Solodov, *A class of inexact variable metric proximal point algorithms*, SIAM J. Optim., 19 (2008), pp. 240–260.
[37] L. Qi and X. Chen, *A preconditioning proximal Newton's method for nondifferentiable convex optimization*, Math. Program., 76 (1995), pp. 411–430.
[38] S. M. Robinson, *An Implicit-Function Theorem for Generalized Variational Inequalities*, Technical summary report 1672, Mathematics Research Center, University of Wisconsin-Madison, 1976.
[39] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, in Mathematical Programming at Oberwolfach, Math. Program. Stud., Springer, New York, 1981, pp. 206–214.
[40] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[41] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.
[42] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.
[43] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer, New York, 2009.
[44] Y. Saad, *Iterative Methods*, PWS Publishing, Boston, 1996.
[45] L. Schork and J. Gondzio, *Implementation of an Interior Point Method with Basis Preconditioning*, Math. Program. Comput., 2020, https://doi.org/10.1007/s12532-020-00181-8.
[46] D. F. Sun, J. Y. Han, and Y. Zhao, *On the finite termination of the damped-newton algorithm for the linear complementarity problem*, Acta Math. Appl. Sin., 21 (1998), pp. 148–154.
[47] T. Valkonen, *Testing and non-linear preconditioning of the proximal point method*, Appl. Math. Optim., 2018, https://doi.org/10.1007/s00245-018-9541-6.
[48] T. Valkonen, *Preconditioned Proximal Point Methods and Notions of Partial Subregularity*, preprint, https://arxiv.org/abs/1711.05123.
[49] N. Veldt, A. Wirth, and D. Gleich, *Correlation clustering with low-rank matrices*, in Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1025–1034.
[50] S. J. Wright, *Implementing proximal point methods for linear programming*, J. Optim. Theory Appl., 65 (1990), pp. 531–554.
[51] L. Q. Yang, D. F. Sun, and K.-C. Toh, *SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints*, Math. Program. Comput., 7 (2015), pp. 331–366.
[52] E.-H. Yen, K. Zhong, C.-J. Hsieh, P. K. Ravikumar, and I. S. Dhillon, *Sparse linear programming via primal and dual augmented coordinate descent*, in Proceedings of NIPS, 2015, pp. 2368–2376.
[53] Y. J. Zhang, N. Zhang, D. F. Sun, and K.-C. Toh, *An efficient Hessian based algorithm for solving large-scale sparse group Lasso problems*, Math. Program., 179 (2020), pp. 223–263.
[54] X.-Y. Zhao, D. F. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2010), pp. 1737–1765.