A parallelizable augmented Lagrangian method applied to large-scale non-convex-constrained optimization problems

Boland, Natashia; Christiansen, Jeffrey; Dandurand, Brian; Eberhard, Andrew; Oliveira, Fabricio

Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., & Oliveira, F. (2019). A parallelizable augmented Lagrangian method applied to large-scale non-convex-constrained optimization problems. Mathematical Programming, Series A, 175(1–2), 503–536. https://doi.org/10.1007/s10107-018-1253-9
Document Version: Accepted Manuscript

Published Version: https://doi.org/10.1007/s10107-018-1253-9

Noname manuscript No.
(will be inserted by the editor)

A parallelizable augmented Lagrangian method applied to large-scale non-convex-constrained optimization problems

Natashia Boland · Jeffrey Christiansen · Brian Dandurand · Andrew Eberhard · Fabricio Oliveira

Received: date / Accepted: date

Abstract We contribute improvements to a Lagrangian dual solution approach applied to large-scale optimization problems whose objective functions are convex, continuously differentiable and possibly nonlinear, while the non-relaxed constraint set is compact but not necessarily convex. Such problems arise, for example, in the split-variable deterministic reformulation of stochastic mixed-integer optimization problems. We adapt the augmented Lagrangian method framework to address the presence of nonconvexity in the non-relaxed constraint set and to enable efficient parallelization. The development of our approach is most naturally compared with the development of proximal bundle methods and especially with their use of serious step conditions. However, deviations from these developments allow for an improvement in efficiency with which parallelization can be utilized. Pivotal in our modification to the augmented Lagrangian method is an integration of the simplicial decomposition method (SDM) and the nonlinear block Gauss-Seidel (GS) method. An adaptation of a serious step condition associated with proximal bundle methods

This work was supported by the Australian Research Council (ARC) grant ARC DP140100985.

N. Boland
Georgia Institute of Technology, Atlanta, Georgia, USA

J. Christiansen
RMIT University, Melbourne, Victoria, Australia

B. Dandurand
RMIT University, Melbourne, Victoria, Australia

A. Eberhard
RMIT University, Melbourne, Victoria, Australia
Tel.: +61-3-9925-2616
Fax: +61-3-9925-1748
E-mail: andy.eberhard@rmit.edu.au

F. Oliveira
Aalto University, Espoo, Finland

allows for the approximation tolerance to be automatically adjusted. Under mild conditions optimal dual convergence is proven, and we report computational results on test instances from the stochastic optimization literature. We demonstrate improvement in parallel speedup over a baseline parallel approach.

Keywords augmented Lagrangian method · proximal bundle method · nonlinear block Gauss-Seidel method · simplicial decomposition method · parallel computing

Mathematics Subject Classification (2000) 90-08, 90C06, 90C11, 90C15, 90C25, 90C26, 90C30, 90C46

1 Introduction

We develop a dual solution approach to the problem of interest having the form

$$\zeta^* := \min_{x,z} \{f(x) : Qx = z, x \in X, z \in Z\}, \tag{1}$$

where f is convex and continuously differentiable, $Q \in \mathbb{R}^{q \times n}$ is a block-diagonal matrix determining linear constraints $Qx = z$, $X \subset \mathbb{R}^n$ is a closed and bounded set, and $Z \subset \mathbb{R}^q$ is a linear subspace. The vector $x \in X$ of decision variables is derived from the original decisions associated with a problem, while the vector $z \in Z$ of auxiliary variables are introduced to effect a decomposable structure in (1). In particular, the decomposable structure takes the form: 1) $X = \prod_{i=1}^m X_i$ with $X_i \subset \mathbb{R}^{n_i}$ closed and bounded and $\sum_{i=1}^m n_i = n$; 2) $f(x) = \sum_{i=1}^m f_i(x_i)$ where $f_i : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ are convex and differentiable for $i = 1, \ldots, m$; 3) Q has block diagonal structure with block diagonal components denoted as $Q_i \in \mathbb{R}^{q_i \times n_i}$, $i = 1, \ldots, m$ where $\sum_{i=1}^m q_i = q$, so that after setting $z = (z_i)_{i=1,\ldots,m}$, where for each $i = 1, \ldots, m$, $z_i \in \mathbb{R}^{q_i}$, we may write $Qx = z$ as $Q_i x_i = z_i$, $i = 1, \ldots, m$. This decomposable structure is implicitly present throughout the development of this paper, although explicit referral to it is typically avoided where it is not needed. We make no other explicit assumptions on f, Q, X, or Z, but we otherwise assume that problem (1) is feasible with finite optimal value.

Problem (1) is general enough to subsume the split-variable deterministic reformulation of a stochastic optimization problem with potentially multiple stages, as defined, for example, in [8], while it can also model the case where f is nonlinear and convex and/or X is any compact (but not necessarily convex) set.

We denote the convex hull of X by $\text{conv}(X)$. The nonconvexity of X is avoided in our development in the sense that 1) an oracle is assumed to be provided for solving a subproblem with linear objective over feasible set X, and 2) another oracle is assumed to be provided for solving a subproblem with convex (possibly nonlinear) objective over a closed, convex, and nonempty subset of $\text{conv}(X)$.

We develop a solution approach to solving the following relaxation of (1),

$$\zeta^{CLD} := \min_{x,z} \{f(x) : Qx = z, x \in \text{conv}(X), z \in Z\} \tag{2}$$

and its Lagrangian dual problem due to the relaxation of $Qx = z$,

$$\zeta^{CLD} = \sup_{\omega} \phi^C(\omega), \tag{3}$$

which is based on the dual function

$$\phi^C(\omega) := \min_{x} \left\{ f(x) + \omega^\top (Qx - z) : x \in \text{conv}(X),\ z \in Z \right\}. \tag{4}$$

When f is linear, then $\phi^C(\omega) = \phi(\omega)$ where

$$\phi(\omega) := \min_{x,z} \left\{ f(x) + \omega^\top (Qx - z), x \in X, z \in Z \right\}. \tag{5}$$

(That is, when f is linear, the role of X and $\text{conv}(X)$ are interchangeable.) Consequently, when f is linear, $\zeta^{CLD} = \zeta^{LD} := \sup_\omega \phi(\omega)$. However, when f is nonlinear, then in general, $\zeta^{CLD} \leqslant \zeta^{LD}$. Strict inequality is demonstrated with the following example. Let $f : \mathbb{R}^2 \mapsto \mathbb{R}$ be defined by $f(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$, $X = \{0,1\} \times \{0,1\}$, and let $Qx = z$ be defined to model the constraints $x_1 - z_1 = 0$ and $x_2 - z_2 = 0$ where $Z = \{(z_1, z_2) : z_1 = z_2\} \subset \mathbb{R}^2$. We see trivially that $\zeta^{CLD} = 0$, which is verified with the saddle point $x_1^* = x_2^* = z_1^* = z_2^* = 0.5$ and $\omega^* = (0,0)$. However, $\zeta^{LD} = 0.5$, which is verified with either of the saddle points $x_1^* = x_2^* = z_1^* = z_2^* = 0$ and $\omega^* = (0,0)$, or $x_1^* = x_2^* = z_1^* = z_2^* = 1$ and $\omega^* = (0,0)$. Thus, $\zeta^{CLD} < \zeta^{LD}$.

Given that X is compact and f is continuous, in order for $-\infty < \phi^C(\omega)$ to hold, it is necessary and sufficient that the dual feasibility assumption

$$\omega \in Z^\perp := \left\{ v \in \mathbb{R}^q : v^\top z = 0 \text{ for all } z \in Z \right\} \tag{6}$$

is maintained either by assumption or by construction. Under condition (6), the z term in definition (4) vanishes, and we may compute

$$\phi^C(\omega) = \min_{x} \left\{ f(x) + \omega^\top Qx : x \in \text{conv}(X) \right\}.$$

Consequently, ϕ^C becomes separable as

$$\phi^C(\omega) = \sum_{i=1}^{m} \phi_i^C(\omega_i),$$

where $\phi_i^C(\omega_i) := \min_x \left\{ f_i(x_i) + \omega_i^\top Q_i x_i : x_i \in \text{conv}(X_i) \right\}$ and $\omega = (\omega_1, \ldots, \omega_m) \in \mathbb{R}^{q_1} \times \cdots \times \mathbb{R}^{q_m}$ has a block structure compatible with the block diagonal structure of Q.

Given that X is closed and bounded (thus so is $\text{conv}(X)$), and (2) is assumed to be feasible, then in order to guarantee that the maximum in (3) is realized for some $\omega^* \in Z^\perp$, we assume a constraint qualification such as Slater's condition. In other words, we assume that there exists (x^*, z^*) such

that $x^* \in \text{int}(\text{conv}(X))$ and $Qx^* = z^*$, where $\text{int}(\cdot)$ returns the topological interior of the set argument. If $\text{conv}(X)$ is polyhedral, then even this Slater's condition is not required.

For the Lagrangian dual problem (3), we note that the objective function ϕ^C is concave, even when f is not convex. We can apply a subgradient method (see e.g. [56]; in textbooks [5,54]) for solving (3) in an efficiently parallelizable manner. Such an approach is proposed in [13]. However, it is preferable to make use of structural features of (3) that allow for smoothing or regularization, so that better convergence properties are realized. For this reason, we consider alternative developments based on proximal point methods that are modified to address both of the above two challenges.

In this paper, we develop an iterative solution approach to solving problems (2) and (3) subject to the two main challenges. First, the set X is not convex (for example, it may have mixed-integer constraints as part of its definition), $\text{conv}(X)$ is not given explicitly, and thus, the augmented Lagrangian method is not theoretically supported with constraint set X, or implementable[1] with constraint set $\text{conv}(X)$. Second, the solution approach should be amenable to efficient parallel computation, in the sense of maximizing the computational work that can be parallelized, the memory usage that can be distributed, and minimizing the amount of parallel communication.

The remainder of the paper is organized as follows. In Section 2, we provide a background and literature review. In Section 3, a general algorithmic framework based on the AL method with approximate subproblem solutions is developed and analyzed. In Section 4, a specific implementation of the Section 3 framework is posed based on the integration of SDM and GS methods, which addresses the aforementioned issues of implementability and efficiency of parallelization. In Section 5, computational experiments and their outcomes are described and interpreted. And at last, Section 6 concludes the paper and provides avenues for future work.

2 Background

As a starting point, we first consider the classical augmented Lagrangian method based on proximal point methods. The augmented Lagrangian (AL) method (also known as the method of multipliers) is developed from proximal point methods, and references include [32,50,4,5].

The AL method typically has favorable convergence properties as a dual solution approach for convex problems (linear convergence rate under certain assumptions, see [52,4] and references cited therein). However, two issues arise in the setting we consider: 1) the set X is not convex, and so current theories of convergence are not applicable; and 2) the primal subproblem associated with each iteration of the AL method is not separable due to the augmented Lagrange term, making efficient parallel implementations difficult to develop.

[1] Implementable is used in the sense of ability to implement the required functionality of an algorithm, not the meaning of implementable specific to stochastic optimization.

The proximal bundle method initially appeared in [39], and for a survey, which also gives its history, see [48]. Use of inexact oracles for computing $\phi(\omega)$ and elements of the subdifferential set $\partial\phi(\omega)$ are studied in [48,49,29] and references therein. In its dual form, the bundle method may be referred to as the stabilized column generation method [3] or the proximal simplicial decomposition method [7]. In implementation, the algorithm we develop here more closely resembles the latter dual form.

For parallelization of the proximal bundle method, see [22] and [41]. The approach developed in this paper is most naturally compared with [41], as both approaches address the manner in which the same continuous master problem is approximately solved. The approach of [22] uses a substantially different parallel computational paradigm based on subspace optimization. This approach, in which solution subspaces are assigned to processors based on periodically updated global state information, is not necessarily based on the problem's decomposable structure.

The proximal bundle method approach requires modification for efficient parallelization. This matter is addressed in [41], where a solution to the continuous master problem is obtained by primal dual interior point methods that exploit the decomposable structure present in the augmented Lagrangian term. We provide and analyze an alternative approach based on the use of:

1. the simplicial decomposition method (SDM) [34,59,5,6], which provides an alternative framework to the proximal bundle method to address the implementability of the proximal point method while allowing for the possibility that f is nonlinear; and
2. nonlinear block Gauss-Seidel (GS) method [33,60,27,58,11] to approximate the solutions to the continuous master problem.

Motivated by its constituent parts, the algorithm we develop is referred to as SDM-GS-ALM.

In an iteration of SDM-GS-ALM, the analog to the continuous master problem is not solved to (near) exactness; instead, approximate solutions based on possibly just one nonlinear block GS iteration are used. Due to the underlying need for convexification of the non-relaxed constraint set, implementability requires that the nonlinear block GS method must be integrated with the SDM so that optimal convergence of the resulting iterations can be established. In this way, a serious step condition similar to that used in proximal bundle methods is eventually satisfied after a finite number of such integrated SDM-GS iterations, and analogous dual optimal convergence of our approach is recovered even with the deviations from the proximal bundle method. In summary, we algorithmically integrate the AL method, the SDM, nonlinear block GS iterations, and the proximal bundle method serious step condition. A convergence analysis is also provided for SDM-GS-ALM. Such an integration allows for a considerable improvement in parallel efficiency with respect to maximizing the computational work that can be parallelized, the memory usage that can be distributed, and minimizing the amount of parallel communication.

Other methods developed in the past that are related to aspects of our contribution include the following. In terms of approximating within the AL method, we include reference to [17,19], where the research goal of developing implementable approximation criteria is addressed. The separable augmented Lagrangian (SALA) method [28], which is an application of the alternating direction method of multipliers (ADMM) [26,23,12] with a form of resource allocation decomposition and incorporates separability into the AL method. Other approaches to introducing separability into the AL method include [14,57]. Jacobi iterate approaches applied within either a proximal bundle method or an AL method framework are considered in [45,55]; the accelerated distributed augmented Lagrangian method (ADAL) developed in [14] is like a Jacobi-iterate analogue of ADMM with supporting convergence analysis. Other approaches to incorporating separability are found in the alternating linearization approaches [37,40] and the predictor corrector proximal multiplier (PCPM) methods [15,31]. All of these methods provide implementable mechanisms for approximating primal subproblem solutions and effecting parallelism in a setting where X is convex. However, they are not practically implementable in our setting where X is not convex and its convex hull $\text{conv}(X)$ is not given beforehand in a computationally useful closed-form description.

Another recently developed algorithm, referred to as FW-PH [10], is closely related to the SDM-GS-ALM algorithm developed in this paper. In terms of functionality, both appear as modifications to ADMM with inner approximated subproblem solutions. While the algorithms differ only slightly in terms of functionality, there are substantial differences in the motivation and the convergence analysis. The convergence analysis of FW-PH interfaces with the convergence analysis for ADMM, which is most naturally developed in the context of the theory of maximal monotone operators and Douglas-Rachford splitting methods [18,20], or as the proximal decomposition of the graph of a maximal monotone operator [43]. In contrast, the convergence analysis of SDM-GS-ALM naturally reflects its synthesis of SDM, the nonlinear block GS method, the proximal bundle method, and the AL method. The convergence analysis of SDM-GS-ALM follows under more general assumptions than that for FW-PH. In particular, the convergence analysis of SDM-GS-ALM allows for trimming of the inner approximations, and it does not require the warm-starting required by FW-PH. The most important difference in functionality is due to the influence of ideas from proximal bundle methods in SDM-GS-ALM, where updates of ω are taken conditionally at each iteration, while such updates are taken unconditionally at each iteration of FW-PH. We shall see that these conditional updates help to mitigate performance problems that arise due to the seemingly inevitable use of suboptimal algorithm parameters.

In papers such as [24,21], ADMM is applied directly to the primal problem (1). In both works, it is acknowledged that ADMM is not theoretically supported in optimal convergence due to the lack of convexity of X. Nevertheless, [24] reports the potential for Lagrangian dual bounds to be recovered at each iteration of ADMM even though it is applied to (1). In [21], where ADMM is applied to nonconvex decentralized unit commitment problems, heuristic im-

provements to ADMM are introduced to address the lack of convexity due to the mixed-integer constraints. In contrast to both of these approaches, where ADMM is applied directly to the primal problem (1), the approach developed in this paper, and its related approach [10], both resemble ADMM but with application to a primal characterization of the dual problem. In these two approaches, the challenge of not having an explicit form for this primal characterization is addressed.

3 An alternative AL approximation approach

Algorithm 1 provides a general framework for an AL method with approximate subproblem solutions that uses the bundle method's serious step condition (SSC). This framework will be useful to guide the developments presented next, in which we discuss optimality conditions and convergence properties of the algorithm. The convergence proof of Algorithm 1 is based on the convergence proofs of the proximal bundle method such as found in Chapter 7 of [54]. In the following we denote the augmented Lagrangian (AL):

$$\begin{aligned} L_\rho(x, z, \omega) &:= f(x) + \omega^\top (Qx - z) + \frac{\rho}{2}\|Qx - z\|^2 \\ &= f(x) + \omega^\top Qx + \frac{\rho}{2}\|Qx - z\|^2 \end{aligned} \tag{7}$$

where the AL relaxes $Qx = z$ and with the second equality following from $\omega \in Z^\perp$ and $z \in Z$.

In the proximal bundle method, the dual function ϕ is approximated by a cutting plane model function that majorizes ϕ. Instead we use the following approximation $\widehat{\phi} : \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}^q \mapsto \mathbb{R}$ of ϕ^C centered at (x^k, z^k), $k \geqslant 0$, to replace the cutting plane model:

$$\widehat{\phi}(\omega, x^k, z^k) := L_\rho(x^k, z^k, \omega) + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2.$$

The convex hull $\mathrm{conv}(X)$ is not known explicitly, and so ϕ^C cannot be evaluated directly. Consequently, we additionally make use of the following minorization $\check{\phi}$ of ϕ^C that can be evaluated. For $x^k \in \mathrm{conv}(X)$, $k \geqslant 0$, define $\check{\phi}(\omega, x^k)$ as follows:

$$\check{\phi}(\omega, x^k) := \min_x \left\{ f(x^k) + \nabla_x f(x^k)^\top (x - x^k) + \omega^\top Qx : x \in X \right\}. \tag{8}$$

In Algorithm 1, a proximal bundle method-like serious step condition is used in Line 9 that makes use of $\widehat{\phi}$ and $\check{\phi}$ in place of the cutting plane model and ϕ, respectively. The inputs f, Q, X, and Z specify the data associated with problem (1); $\rho > 0$ is the AL term coefficient; ω^0 is an initial dual solution; $\gamma \in (0, 1)$ is the parameter of the serious step condition of Line 9; and $\epsilon > 0$ is a tolerance for termination. Algorithm 1 will be given a specific implementation in the form of SDM-GS-ALM in Section 4.

Algorithm 1 A general approximated ALM using a bundle method SSC.

Preconditions: $\omega^1 \in Z^\perp$, $\gamma \in (0,1)$.
function ApproxALM(f, Q, X, Z, ρ, ω^1, γ, ϵ, k_{max})
  for $k = 1, 2, \ldots, k_{max}$ **do**
    Solve approximately
      $(x^k, z^k) \in \operatorname{argmin}_{x,z} \left\{ L_\rho(x, z, \omega^k) : x \in \operatorname{conv}(X), z \in Z \right\}$ such that
        1) $z^k \in \operatorname{argmin}_z \left\{ \|Qx^k - z\|_2^2 : z \in Z \right\}$ and
        2) either
          $\widehat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1}) \leqslant \epsilon$ or
          $0 < \gamma \leqslant \frac{\check{\phi}\left(\omega^k + \rho(Qx^k - z^k), x^k\right) - \check{\phi}(\omega^k, x^{k-1})}{\widehat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1})}$
    if $\widehat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1}) \leqslant \epsilon$ **then**
      return (x^k, z^k, ω^k)
    else
      set $\omega^{k+1} \leftarrow \omega^k + \rho(Qx^k - z^k)$
    end if
  end for
  return (x^k, z^k, ω^{k+1})
end function

Remark 1 Under the assumption that $\operatorname{conv}(X)$ is not known beforehand by any characterization, direct evaluation of ϕ^C or any of its subgradients at any $\omega \in Z^\perp$ is not possible. This dual function is not used in the proximal bundle method and is only treated indirectly in the current development.

In addition to generating a sequence $\{\omega^k\}$ of dual solutions to (3), our algorithm will also generate a sequence of primal solutions $\{(x^k, z^k)\}$ to (2), and so reference to (2) will be useful. In applying the AL method to problem (2), the continuous master problem for fixed $\omega \in Z^\perp$ takes the form

$$\zeta_\rho^{AL}(\omega) := \min_{x,z} \{ L_\rho(x, z, \omega), x \in \operatorname{conv}(X), z \in Z \}. \tag{9}$$

Lemma 1 *For any optimal solution ω^* to problem (3), we have $\zeta_\rho^{AL}(\omega^*) = \zeta^{CLD}$. Additionally, any optimal solution (x^*, z^*) to problem (9) with $\omega = \omega^*$ is also optimal for problem (2).*

Proof We specialize developments in, e.g., Section 4 of [53] or Section 6.4.3 of [54]. In the following, we begin assuming $\overline{\omega} \in Z^\perp$ as an arbitrary fixed vector to show:

$$\max_{\omega \in Z^\perp} \left\{ (\omega - \overline{\omega})^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 \right\} = \frac{\rho}{2} \min_{z \in Z} \|Qx - z\|_2^2. \tag{10}$$

By the uniqueness of the projection onto a subspace, $z \in \operatorname{argmin}_{z \in Z} \|Qx - z\|_2^2$ is the unique $z \in Z$ for which $(Qx - z) \in Z^\perp$. Moreover, the optimality condition for left-hand side problem in (10): $0 \in \partial_\omega \left\{ (\omega - \overline{\omega})^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 + \delta_{Z^\perp}(\omega) \right\}$ dictate that $(\omega - \overline{\omega}) \in \rho(Qx + Z)$ (where we have used, for $\omega \in Z^\perp$, $\partial \delta_{Z^\perp}(\omega) = N_{Z^\perp}(\omega) = Z$, where $\delta_{Z^\perp}$ denotes the indicator function of a convex set $Z^\perp$ and

$N_{Z^\perp}(\omega)$ the normal cone at ω). Hence $(\omega - \overline{\omega}) = \rho(Qx - z)$ for some $z \in Z$. Furthermore, $\omega - \overline{\omega} \in Z^\perp$ must hold, and so $z \in Z$ must be chosen so that $\rho(Qx - z) \in Z^\perp$ also. Consequently, from our first observation, this $z \in Z$ must be the unique solution of the right-hand side of (10). Evaluating the objective on the left-hand side of (10) (observing that we must have $\rho(Qx - z) \in Z^\perp$) establishes the claimed equality. We may compute:

$$\begin{aligned}
&\max_{\omega \in Z^\perp} \phi^C(\omega) - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 \qquad (11)\\
&= \max_{\omega \in Z^\perp} \min_x \left\{ f(x) + \omega^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 : x \in \mathrm{conv}(X) \right\}\\
&= \min_x \left\{ \begin{array}{l} f(x) + \overline{\omega}^\top Qx \\ + \max_{\omega \in Z^\perp} \left\{ (\omega - \overline{\omega})^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 \right\} \end{array} : x \in \mathrm{conv}(X) \right\}\\
&= \min_x \left\{ f(x) + \overline{\omega}^\top Qx + \frac{\rho}{2} \min_{z \in Z} \left\{ \|Qx - z\|_2^2 \right\} : x \in \mathrm{conv}(X) \right\}\\
&= \min_{x,z} \left\{ L_\rho(x, z, \overline{\omega}), x \in \mathrm{conv}(X), z \in Z \right\}.
\end{aligned}$$

The switching of min and max is justified by the Sion min-max theorem along with the convexity of f, $\mathrm{conv}(X)$, Z and $\mathrm{conv}(X)$ assumed compactness. In substituting $\overline{\omega} = \omega^*$, the value of the left-hand side maximization problem (11) is clearly ζ^{CLD}, while the same substitution on the right-hand side (10) yields the value $\zeta_\rho^{AL}(\omega^*)$, from which we see that $\zeta^{CLD} = \zeta_\rho^{AL}(\omega^*)$. To prove the last claim, we note that $L_\rho(x^*, z^*, \omega^*) = \zeta^{CLD}$ implies that $\|Qx^* - z^*\|_2^2 = 0$. Otherwise, $\phi^C(\omega^*) < \zeta^{CLD}$, contradicting the dual optimality of ω^*. Thus, (x^*, z^*) is feasible and optimal for problem (2).

The approximation $\widehat{\phi}$ satisfies the following bounding relationship.

Lemma 2 *For each (x^k, z^k), $k \geqslant 0$, such that the z-optimality condition is satisfied:*

$$z^k \in \operatorname*{argmin}_z \left\{ \|Qx^k - z\|_2^2 : z \in Z \right\}, \qquad (12)$$

we have for each $\omega \in Z^\perp$

$$\widehat{\phi}(\omega, x^k, z^k) \geqslant \phi^C\left(\omega + \rho(Qx^k - z^k)\right). \qquad (13)$$

Proof Via convexity of the term $\|Qx - z\|_2^2$ over $(x, z) \in \mathrm{conv}(X) \times Z$, we may write the following inequalities that hold for $(x, z) \in \mathrm{conv}(X) \times Z$ and a fixed

$\omega \in Z^\perp$. Via the subgradient inequality:

$$\begin{aligned} L_\rho(x,z,\omega) &\geqslant f(x) + \omega^\top Qx + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2 \\ &\quad + \rho(Qx^k - z^k)^\top (Qx - z) - \rho(Qx^k - z^k)^\top (Qx^k - z^k) \\ &= f(x) + \omega^\top Qx - \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2 + \rho(Qx^k - z^k)^\top (Qx - z) \\ \implies L_\rho(x,z,\omega) + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2 &\geqslant f(x) + \left[\omega + \rho(Qx^k - z^k)\right]^\top Qx \qquad (14) \\ &\geqslant \min_x \left\{ f(x) + \left(\omega + \rho(Qx^k - z^k)\right)^\top Qx : x \in \mathrm{conv}(X) \right\}. \qquad (15) \end{aligned}$$

Note that the term $-\rho(Qx^k - z^k)^\top z$ vanishes due to the optimality condition associated with (12). Inequality (13) follows from the inequalities (14)–(15) once the substitution $(x,z) = (x^k, z^k)$ and the definition of $\widehat{\phi}(\omega, x^k, z^k)$ are applied to the left-hand side of (14).

Observe that, due to the linearity of the objective function with respect to x in (8), the use of constraint sets X and $\mathrm{conv}(X)$ are interchangeable, and so in evaluating $\breve{\phi}$, an explicit description of $\mathrm{conv}(X)$ is not required. Furthermore, from the definition of ϕ^C, the convexity of f over $\mathbb{R}^n$, and the interchangeability of X and $\mathrm{conv}(X)$ in (8), it is clear that for all $x^k \in \mathbb{R}^n$, $k \geqslant 0$, we have $\phi^C(\omega) \geqslant \breve{\phi}(\omega, x^k)$. This is not only true in principle but also practically if one can provide a oracle that returns an extremal point of $\mathrm{conv}(X)$ when minimizing a linear function over X. When the non-convexity is entirely due to the presence of integer restrictions on variables, MIP or MINLP solvers can provide such an oracle. Later in section 4 we shall see that the class of problems that is amenable to the final implementable algorithm is dictated, in practice, by the user's ability to provide such an oracle. Moreover, when f is linear, we have $\phi^C(\omega) \equiv \breve{\phi}(\omega, x^k)$ for all x^k, $k \geqslant 0$; the two functions collapse into the same function with the centering at x^k of the latter function now irrelevant.

The first important property of $(\omega, x) \mapsto \breve{\phi}(\omega, x)$ is its continuity.

Lemma 3 *Let X be compact, and f be continuously differentiable. Then $(\overline{\omega}, \overline{x}) \mapsto \breve{\phi}(\overline{\omega}, \overline{x})$ is continuous over $(\overline{\omega}, \overline{x}) \in Z^\perp \times \mathbb{R}^n$.*

Proof From (8), compute

$$\begin{aligned} \breve{\phi}(\overline{\omega}, \overline{x}) &= f(\overline{x}) - \nabla_x f(\overline{x})^\top \overline{x} + \min_x \left\{ \left[\nabla_x f(\overline{x}) + \overline{\omega}^\top Q\right] x + \delta_{\mathrm{conv}(X)}(x) \right\} \\ &= f(\overline{x}) - \nabla_x f(\overline{x})^\top \overline{x} - \delta^*_{\mathrm{conv}(X)}\left(-\left[\nabla_x f(\overline{x}) + \overline{\omega}^\top Q\right]\right). \end{aligned}$$

where $\delta_{\mathrm{conv}(X)}(x) := \begin{cases} 0, & \text{if } x \in \mathrm{conv}(X); \\ \infty, & \text{otherwise.} \end{cases}$ is the indicator function on the set $\mathrm{conv}(X)$ and $\delta^*_{\mathrm{conv}(X)}$ is the conjugate function [51] of $\delta_{\mathrm{conv}(X)}$. As $\mathrm{conv}(X)$ is convex and compact, we see that $\delta^*_{\mathrm{conv}(X)}(\cdot)$ has domain $\mathbb{R}^n$ and is thus continuous over $\mathbb{R}^n$ (e.g., Lemma 2.91 of [54]), yielding the intended conclusion.

The second property of $\breve{\phi}$ is its limiting behavior as the solutions (x^k, z^k) approach certain critical values.

Lemma 4 *Let the sequence* $\{(x^k, z^k)\} \subset \mathrm{conv}(X) \times Z$ *satisfy the z-optimality condition* (12) *for each* $k \geqslant 1$. *If, for some fixed* $\omega \in Z^{\perp}$, *the sequence* $\{(x^k, z^k)\}$ *converges optimally in the sense that*

$$\lim_{k\to\infty} (x^k, z^k) = (x^*, z^*) \in \operatorname*{argmin}_{x,z} \{L_\rho(x, z, \omega) : x \in \mathrm{conv}(X), z \in Z\},$$

then

$$\lim_{k\to\infty} \breve{\phi}(\omega + \rho(Qx^k - z^k), x^k) = L_\rho(x^*, z^*, \omega) + \frac{\rho}{2} \|Qx^* - z^*\|_2^2. \tag{16}$$

Proof We begin by writing the necessary (and sufficient) conditions associated with the optimality $(x^*, z^*) \in \mathrm{argmin}_{x,z} \{L_\rho(x, z, \omega) : x \in \mathrm{conv}(X), z \in Z\}$:

$$\begin{bmatrix} \nabla f(x^*) + [\omega + \rho(Qx^* - z^*)]^\top Q \\ -\rho(Qx^* - z^*) \end{bmatrix} \begin{bmatrix} x - x^* \\ z - z^* \end{bmatrix} \geqslant 0 \quad \text{for all } x \in \mathrm{conv}(X), z \in Z.$$

Since $z^k \in \mathrm{argmin}_z \left\{ \|Qx^k - z\|_2^2 : z \in Z \right\}$ for each $k \geqslant 1$, we have $Qx^k - z^k \in Z^{\perp}$, and so $Qx^* - z^* \in Z^{\perp}$ also. Thus, we can simplify the consideration of the above displayed necessary conditions to consider the x block only:

$$\left[\nabla f(x^*) + [\omega + \rho(Qx^* - z^*)]^\top Q \right]^\top \left[x - x^* \right] \geqslant 0 \quad \text{for all } x \in \mathrm{conv}(X),$$

which implies

$$\min_x \left\{ \left[\nabla f(x^*) + [\omega + \rho(Qx^* - z^*)]^\top Q \right]^\top \left[x - x^* \right] : x \in \mathrm{conv}(X) \right\} = 0.$$

In terms of $\breve{\phi}(\omega + \rho(Qx^* - z^*), x^*)$, the above equality is re-written as:

$$\begin{aligned} \breve{\phi}(\omega + \rho(Qx^* - z^*), x^*) &= f(x^*) + \omega^\top Qx^* + \rho \|Qx^* - z^*\|_2^2 \\ &= L_\rho(x^*, z^*, \omega) + \frac{\rho}{2} \|Qx^* - z^*\|_2^2, \end{aligned}$$

where the equality $(Qx^* - z^*)^\top z^* = 0$ is utilized. The continuity of $(\overline{\omega}, \overline{x}) \mapsto \breve{\phi}(\overline{\omega}, \overline{x})$ established in Lemma 3 gives the desired conclusion.

We use Lemmas 3 and 4 to develop the proximal bundle method-like serious step condition (SSC) that makes use of $\widehat{\phi}$ and $\breve{\phi}$ in place of the cutting plane model and ϕ, respectively. Defining $\widetilde{\omega}^k := \omega^k + \rho(Qx^k - z^k)$, consider the following modified serious step condition:

$$\gamma \leqslant \frac{\breve{\phi}(\widetilde{\omega}^k, x^k) - \breve{\phi}(\omega^k, x^{k-1})}{\widehat{\phi}(\omega^k, x^k, z^k) - \breve{\phi}(\omega^k, x^{k-1})} \leqslant 1, \tag{17}$$

where $\gamma \in (0, 1)$ is the SSC parameter. The upper bound of (17) is satisfied automatically since $\widehat{\phi}(\omega^k, x^k, z^k) \geqslant \phi^C(\widetilde{\omega}^k) \geqslant \breve{\phi}(\widetilde{\omega}^k, x^k)$ holds by Lemma 2 and the definition of $\breve{\phi}$. However, the satisfaction of the lower bound is conditional on γ.

Remark 2 Throughout this paper, we shall always assume or construct z^k such that the z-optimality condition (12) is satisfied for each $k \geqslant 0$. Due to the necessary conditions of optimality associated with (12) and that Z is a linear subspace, we have $(Qx^k - z^k)^\top z = 0$ for all $z \in Z$. It immediately follows that if $\omega^k \in Z^\perp$, then $\widetilde{\omega}^k = \omega^k + \rho(Qx^k - z^k) \in Z^\perp$ also. Thus, the satisfaction of the z-optimality condition (12) guides the generation of $\{\omega^k\}$ so that if $\omega^0 \in Z^\perp$, then $\omega^k \in Z^\perp$ is always maintained for each $k \geqslant 1$.

Under certain circumstances, the denominator of the ratio displayed in (17) can be zero. The following lemma states that this never happens when ω^k is *not* dual optimal with respect to the dual problem (3).

Lemma 5 *For any $\omega \in Z^\perp$ that is not dual optimal with respect to the dual problem* (3) *and $(x, z) \in \mathrm{conv}(X) \times Z$, we have*

$$\widehat{\phi}(\omega, x, z) - \phi^C(\omega) > 0. \tag{18}$$

Consequently, at any iteration k, the denominator of the ratio displayed in (17) *cannot be zero when ω^k is not dual optimal.*

Proof By the definition of $\widehat{\phi}$, we have

$$\widehat{\phi}(\omega, x, z) - \phi^C(\omega) \geqslant L_\rho(x^*, z^*, \omega) + \frac{\rho}{2}\|Qx - z\|_2^2 - \phi^C(\omega)$$

for all $\mathrm{conv}(X)$ and $z \in Z$, where

$$(x^*, z^*) \in \operatorname*{argmin}_{x,z} \{L_\rho(x, z, \omega) : x \in \mathrm{conv}(X), z \in Z\}.$$

(That is, we substitute $L_\rho(x, z, \omega)$ from the definition of $\widehat{\phi}$ with $L_\rho(x^*, z^*, \omega)$ to get the inequality.) Now $L_\rho(x^*, z^*, \omega) - \phi^C(\omega) > 0$ when ω is not dual optimal. Otherwise, if $L_\rho(x^*, z^*, \omega) = \phi^C(\omega)$, then $Qx^* = z^*$ must hold, and (x^*, z^*, ω) is a Lagrangian saddle point for problem (2) with respect to the Lagrangian relaxation of the constraint $Qx = z$. This contradicts the non-dual optimality of ω. Thus, the strict inequality (18) is established.

In the context of (17) at iteration k, noting that $\phi^C(\omega^k) \geqslant \breve{\phi}(\omega^k, x^{k-1})$, we substitute $(x, z) = (x^k, z^k)$ and $\omega = \omega^k$ in the strict inequality (18) and so the denominator in (17) is positive when ω^k is not dual optimal.

From Lemma 4, we have the following result regarding the satisfaction of condition (17).

Proposition 1 *Let the sequence $\{(x^k, z^k)\} \subset \mathrm{conv}(X) \times Z$ satisfy*

$$z^k \in \operatorname*{argmin}_z \left\{\|Qx^k - z\|_2^2 : z \in Z\right\}$$

for each $k \geqslant 1$. Furthermore, let $\omega \in Z^\perp$ and $\omega \notin \mathrm{argmax}_\omega\, \phi(\omega)$. If the sequence $\{(x^k, z^k)\}$ converges optimally in the sense that

$$\lim_{k\to\infty}(x^k, z^k) = (x^*, z^*) \in \operatorname*{argmin}_{x,z} \{L_\rho(x, z, \omega) : x \in \mathrm{conv}(X), z \in Z\},$$

then condition (17) *must be satisfied after a finite number of iterations.*

Proof For all $(x^k, z^k) \in \text{conv}(X) \times Z$ with $z^k \in \text{argmin}_z \left\{ \left\| Qx^k - z \right\|_2^2 \right\}$, we have

$$\begin{aligned}\widehat{\phi}(\omega, x^k, z^k) &= L_\rho(x^k, z^k, \omega) + \frac{\rho}{2} \left\| Qx^k - z^k \right\|_2^2 \\ &\geqslant \phi^C(\omega + \rho(Qx^k - z^k)) \geqslant \breve{\phi}(\omega + \rho(Qx^k - z^k), x^k),\end{aligned}$$

where the first inequality follows from the definition of $\widehat{\phi}$ and Lemma 2, and the second inequality follows readily from the definition of $\breve{\phi}$, the subgradient inequality and the interchangeability of X and $\text{conv}(X)$. By the assumption that ω is not dual optimal, the denominator of (17) cannot be zero by Lemma 5. It follows from the convergence in (16) implied by Lemma 4 that the ratio in (17) must approach 1, and so condition (17) must be satisfied after a finite number of iterations.

Consequently, unless the current ω^k is already dual optimal, there cannot be an infinite number of null-steps when using condition (17). Recall that we use k to count only serious steps.

Proposition 2 *Assume that problem* (3) *has an optimal dual solution* ω^**, and that for each* $k \geqslant 1$*,* $\phi^C(\omega^k) < \phi^C(\omega^*)$*. Also, assume that* ρ *and* γ *may vary with each iteration, defined by sequences* $\{\rho_k\}$ *and* $\{\gamma_k\}$ *such that* $\rho_k > 0$ *and* $\gamma_k \in (0, 1)$*, bounded strictly away from zero for all* $k \geqslant 1$*, and* $\rho_k \left(\frac{1-\gamma_k}{\gamma_k} \right) = c > 0$ *for all* k*. If the sequence* $\left\{\omega^k\right\}$ *of dual updates is generated with Algorithm 1 with* $\epsilon = 0$ *and* $k_{max} = \infty$*, then* $\left\{\omega^k\right\}$ *converges, and* $\lim_{k\to\infty} \breve{\phi}(\omega^k, x^{k-1}) = \zeta^{CLD}$ *(and consequently* $\lim_{k\to\infty} \phi^C(\omega^k) = \zeta^{CLD}$*). Furthermore,*

$$\lim_{k\to\infty} \widehat{\phi}(\omega^k, x^k, z^k) = \zeta^{CLD},$$

and all limit points $(\bar{x}, \bar{z})$ *of the sequence* $\left\{(x^k, z^k)\right\}$ *are optimal for problem* (2)*.*

Proof Let ω^* be any dual optimal solution for problem (3). For each iteration $k \geqslant 1$, write the following two relations:

$$\begin{aligned}\left\| \omega^{k+1} - \omega^* \right\|_2^2 &= \left\| \omega^k - \omega^* + \rho_k(Qx^k - z^k) \right\|_2^2 \\ &= \left\| \omega^k - \omega^* \right\|_2^2 + 2\rho_k(Qx^k - z^k)^\top(\omega^k - \omega^*) + \rho_k^2 \left\| Qx^k - z^k \right\|_2^2, \end{aligned} \tag{19}$$

$$\begin{aligned}\text{and} \quad \phi^C(\omega^*) &\leqslant L_{\rho_k}(x^k, z^k, \omega^*) = L_{\rho_k}(x^k, z^k, \omega^k) + (\omega^* - \omega^k)^\top(Qx^k - z^k) \\ &\Longrightarrow (\omega^k - \omega^*)^\top(Qx^k - z^k) \leqslant L_{\rho_k}(x^k, z^k, \omega^k) - \phi^C(\omega^*). \end{aligned} \tag{20}$$

Substituting the inequality (20) into equality (19), we have

$$\begin{aligned}\left\|\omega^{k+1}-\omega^*\right\|_2^2 &\leqslant \left\|\omega^k-\omega^*\right\|_2^2 \\ &\quad + 2\rho_k\left[L_{\rho_k}(x^k,z^k,\omega^k)-\phi^C(\omega^*)\right]+\rho_k^2\left\|Qx^k-z^k\right\|_2^2 \qquad (21)\\ &= \left\|\omega^k-\omega^*\right\|_2^2+2\rho_k\left[\breve{\phi}(\omega^k,x^{k-1})-\phi^C(\omega^*)\right] \\ &\quad + 2\rho_k\left[L_{\rho_k}(x^k,z^k,\omega^k)+\frac{\rho_k}{2}\left\|Qx^k-z^k\right\|_2^2-\breve{\phi}(\omega^k,x^{k-1})\right]. \qquad (22)\end{aligned}$$

By assumption, for each $k \geqslant 1$, we have $\phi^C(\omega^k) < \phi^C(\omega^*)$, so by Lemma 5 and $\epsilon = 0$, the Line 8 condition of Algorithm 1 never holds. Thus, the Line 9 condition, which is equivalent to the satisfaction of condition (17), is satisfied for each $k \geqslant 1$. Rewriting (17), with the substitution $\widetilde{\omega}^k = \omega^{k+1}$, as

$$L_{\rho_k}(x^k,z^k,\omega^k)+\frac{\rho_k}{2}\left\|Qx^k-z^k\right\|_2^2-\breve{\phi}(\omega^k,x^{k-1}) \leqslant \frac{\breve{\phi}(\omega^{k+1},x^k)-\breve{\phi}(\omega^k,x^{k-1})}{\gamma_k} \qquad (23)$$

and substituting (23) into (22), we have

$$\begin{aligned}\left\|\omega^{k+1}-\omega^*\right\|_2^2 &\leqslant \left\|\omega^k-\omega^*\right\|_2^2+2\rho_k\left[\breve{\phi}(\omega^k,x^{k-1})-\phi^C(\omega^*)\right] \\ &\quad + \frac{2\rho_k}{\gamma_k}\left[\breve{\phi}(\omega^{k+1},x^k)-\breve{\phi}(\omega^k,x^{k-1})\right] \\ &\leqslant \left\|\omega^k-\omega^*\right\|_2^2+2\rho_k\left[\breve{\phi}(\omega^{k+1},x^k)-\phi^C(\omega^*)\right] \\ &\quad + 2\rho_k\left(\frac{1-\gamma_k}{\gamma_k}\right)\left[\breve{\phi}(\omega^{k+1},x^k)-\breve{\phi}(\omega^k,x^{k-1})\right] \qquad (24)\end{aligned}$$

From (24), we make the following three inferences: 1) that $\left\{\left\|\omega^k-\omega^*\right\|\right\}$ is bounded, 2) that $\sum_{k=1}^{\infty}\left[\phi^C(\omega^*)-\phi^C(\omega^k)\right]$ is finite, and 3) that $\left\{\omega^k\right\}$ converges. To establish these inferences, we rearrange terms and sum the inequality (24) from $k = \ell, \ldots, N$ for some integers $1 \leqslant \ell \leqslant N$ to get

$$\begin{aligned}&2\sum_{k=\ell}^{N}\rho_k\left[\phi^C(\omega^*)-\breve{\phi}(\omega^{k+1},x^k)\right]+\left\|\omega^{N+1}-\omega^*\right\|_2^2 \\ &\quad\leqslant \left\|\omega^{\ell}-\omega^*\right\|_2^2+2\rho_N\left(\frac{1-\gamma_N}{\gamma_N}\right)\breve{\phi}(\omega^{N+1},x^N)-2\rho_\ell\left(\frac{1-\gamma_\ell}{\gamma_\ell}\right)\breve{\phi}(\omega^{\ell},x^{\ell-1}) \\ &\quad + 2\sum_{k=\ell}^{N-1}\left[\rho_k\left(\frac{1-\gamma_k}{\gamma_k}\right)-\rho_{k+1}\left(\frac{1-\gamma_{k+1}}{\gamma_{k+1}}\right)\right]\breve{\phi}(\omega^{k+1},x^k) \\ &\quad\leqslant \left\|\omega^{\ell}-\omega^*\right\|_2^2+2c\left[\phi^C(\omega^*)-\breve{\phi}(\omega^{\ell},x^{\ell-1})\right], \qquad (25)\end{aligned}$$

where the last inequality follows from the assumption that $\rho_k\left(\frac{1-\gamma_k}{\gamma_k}\right) = c$ for all k and the bounding relationships implied by the optimality of ω^*:

$$\phi^C(\omega^*) > \phi^C(\omega^{k+1}) \geqslant \breve{\phi}(\omega^{k+1},x^k) \qquad (26)$$

Noting that each summand $\phi^C(\omega^*) - \breve{\phi}(\omega^{k+1}, x^k)$ in the summation on the left-hand side of (25) is nonnegative so we have immediately from (25) that $\sum_{k=1}^{\infty} \rho_k \left[\phi^C(\omega^*) - \breve{\phi}(\omega^k, x^{k-1})\right] < \infty$ and $\left\{(\omega^k - \omega^*)\right\}$ is bounded, establishing the first two inferences from (24). The validity of the first two inferences imply the boundedness of $\left\{\omega^k\right\}$ and the convergence $\lim_{k\to\infty} \breve{\phi}(\omega^k, x^{k-1}) = \phi^C(\omega^*)$, respectively. The boundedness of $\left\{\omega^k\right\}$ implies the existence of limit points, while the convergence $\lim_{k\to\infty} \breve{\phi}(\omega^k, x^{k-1}) = \phi^C(\omega^*)$ implies that all such limit points are dual optimal. It is straightforward from (26) that $\lim_{k\to\infty} \phi^C(\omega^k) = \phi^C(\omega^*)$ also.

To establish the third assertion, that $\left\{\omega^k\right\}$ in fact converges, we drop the summation from the left-hand side of (25),

$$\left\|\omega^{N+1} - \omega^*\right\|_2^2 \leqslant \left\|\omega^{\ell} - \omega^*\right\|_2^2 + 2c\left[\phi^C(\omega^*) - \breve{\phi}(\omega^{\ell}, x^{\ell-1})\right], \tag{27}$$

and note that the above analysis holds independent of the choice of dual optimal ω^*. Since it was just shown that $\left\{\omega^k\right\}$ has limit points, and that all such limit points are dual optimal, we now specify ω^* to be one of these limit points. We then choose an appropriate ℓ for any $\varepsilon > 0$ so that the right-hand side of (27) is arbitrarily small, i.e.,

$$\left\|\omega^{N+1} - \omega^*\right\|_2^2 \leqslant \varepsilon$$

for all $N \geqslant \ell$. Thus, $\lim_{k\to\infty} \omega^k = \omega^*$, and it is clear that the limit point ω^* of $\left\{\omega^k\right\}$ is in fact unique.

To prove the last assertion, the satisfaction of (17) is rewritten as

$$\begin{aligned}\breve{\phi}(\omega^{k+1}, x^k) - \breve{\phi}(\omega^k, x^{k-1}) &\leqslant \widehat{\phi}(\omega^k, x^k, z^k) - \breve{\phi}(\omega^k, x^{k-1}) \\ &\leqslant \frac{1}{\gamma_k}\left(\breve{\phi}(\omega^{k+1}, x^k) - \breve{\phi}(\omega^k, x^{k-1})\right).\end{aligned}$$

Due to the convergence $\lim_{k\to\infty} \breve{\phi}(\omega^k, x^{k-1}) = \zeta^{CLD}$, we have on taking the limit as $k \to \infty$ of the last displayed inequalities that $\lim_{k\to\infty} \widehat{\phi}(\omega^k, x^k, z^k) = \zeta^{CLD}$. Noting that $\widehat{\phi}(\omega^k, x^k, z^k) = L_{\rho_k}(x^k, z^k, \omega^k) + \frac{\rho_k}{2}\left\|Qx^k - z^k\right\|_2^2$, it is clear that if $\limsup_{k\to\infty} \rho_k = \infty$, then we have $\lim_{k\to\infty} \left\|Qx^k - z^k\right\|_2^2 = 0$ and $\lim_{k\to\infty} L_{\rho_k}(x^k, z^k, \omega^k) = \zeta^{CLD}$, and so the limit points of $\left\{(x^k, z^k)\right\}$ must be feasible and furthermore optimal for (2). Now assume $0 < \limsup \rho_k < \infty$. In taking the limit points $(\bar{x}, \bar{z}, \omega^*)$ of the sequence $\left\{(x^k, z^k, \omega^k)\right\}$ and $\bar{\rho}$ of $\{\rho_k\}$, noting that the optimal value of problem (9) with $\omega = \omega^*$ is ζ^{CLD} by Lemma 1,

$$\zeta^{CLD} + \frac{\bar{\rho}}{2}\left\|Q\bar{x} - \bar{z}\right\|_2^2 \leqslant L_{\bar{\rho}}(\bar{x}, \bar{z}, \omega^*) + \frac{\bar{\rho}}{2}\left\|Q\bar{x} - \bar{z}\right\|_2^2 = \zeta^{CLD}.$$

From this, it follows that $\left\|Q\bar{x} - \bar{z}\right\|_2^2 = 0$ and $L_{\bar{\rho}}(\bar{x}, \bar{z}, \omega^*) = \zeta^{CLD}$, and so $(\bar{x}, \bar{z})$ must be feasible and furthermore optimal for (2).

3.1 Rate-of-convergence

The proof of Proposition 2 allows for the following remarks on the rate-of-convergence associated with $\lim_{k\to\infty} \breve{\phi}(\omega^k, x^{k-1}) = \zeta^{CLD}$. Note that each iteration k of Algorithm 1 corresponds to a serious step update of ω^k.

1. Let $\rho_k = \rho$ and $\gamma_k = \gamma$ for all $k \geqslant 1$, where $\rho > 0$ and $\gamma \in (0, 1)$ are constants. Then, from the proof of Proposition 2, we have

$$\sum_{k=1}^{N} \left[\phi^C(\omega^*) - \breve{\phi}(\omega^k, x^{k-1}) \right] < \infty$$

   and since $\left\{ \breve{\phi}(\omega^k, x^{k-1}) \right\}$ is monotonically non-decreasing, it is clear that

$$\phi^C(\omega^*) - \breve{\phi}(\omega^k, x^{k-1}) = o(1/k),$$

   where o is the Little-o notation.
2. Let $\rho_k = k\rho$ for some constant $\rho > 0$ and $\gamma_k = \frac{\rho_k}{c+\rho_k}$ for some constant $c > 0$ so that $\rho_k \left(\frac{1-\gamma_k}{\gamma_k} \right) = c$ is a constant. Then we have

$$\begin{aligned} &\textstyle\sum_{k=1}^{N} k \left[\phi^C(\omega^*) - \breve{\phi}(\omega^k, x^{k-1}) \right] < \infty, \\ \text{and so} \qquad &\phi^C(\omega^*) - \breve{\phi}(\omega^k, x^{k-1}) = o(1/k^2). \end{aligned}$$

3. Let $\rho_k = \rho b^k$ for some constant $\rho > 0$ and $b > 1$, and as $\gamma_k = \frac{\rho_k}{c+\rho_k}$ for some constant $c > 0$ so that $\rho_k \left(\frac{1-\gamma_k}{\gamma_k} \right) = c$, a constant. Then we have

$$\begin{aligned} &\textstyle\sum_{k=1}^{N} b^k \left[\phi^C(\omega^*) - \breve{\phi}(\omega^k, x^{k-1}) \right] < \infty, \\ \text{and so} \qquad &\phi^C(\omega^*) - \breve{\phi}(\omega^k, x^{k-1}) = o(b^{-k}). \end{aligned}$$

Since the number of null step updates per serious step is not fixed, and a null step does not require significantly less computational effort, these convergence results in terms of serious steps cannot be generalised to a convergence result in terms of total steps or runtime without some notion of how often null steps are taken.

Exploratory numerical tests did not conclusively reveal any clear and consistent pattern to the frequency of null steps as the algorithm progresses. However, for at least some combinations of instances and parameters the null step frequency increased significantly as the duality gap decreased. Therefore, the practical convergence behaviour of the algorithm in terms of runtime is likely worse than the above results would suggest.

4 Main algorithm

After integrating SDM and the nonlinear block Gauss-Seidel method, a practical implementation of Algorithm 1 is provided in this section.

We consider the following general two-block problem

$$\min_{x,z} \{F(x,z) : x \in \mathrm{conv}(X), z \in Z\} \tag{28}$$

where $F : \mathbb{R}^n \times \mathbb{R}^q \mapsto \mathbb{R}$ is a continuously differentiable function, $\mathrm{conv}(X)$ and Z are closed convex sets, and $\mathrm{conv}(X)$ is also bounded. (Z can be more generally a convex set in this setting, not necessarily a linear (sub)space.) Additionally, we assume for each fixed $x \in \mathrm{conv}(X)$ that $z \mapsto F(x,z)$ is inf-compact. (That is, the set $\{z \in Z : F(x,z) \leqslant \ell\}$ is compact for all $x \in \mathrm{conv}(X)$ and $\ell \in \mathbb{R}$.) In the context of Algorithm 1, we would identify $F(x,z) = L_\rho(x,z,\omega)$ for a given ω.

Problem (28) is assumed to be feasible, bounded, and to have an optimal solution (x^*, z^*). We shall utilize the following two-block nonlinear Gauss-Seidel (GS) method with the x update approximated in a manner resembling an iteration of the SDM. We assume the user provides an oracle to return an extremal point in $\mathrm{conv}(X)$ when minimizing a linear function over X. This can be used to initialize the following and later algorithms.

Algorithm 2 An iteration of inner-approximated nonlinear Gauss-Seidel approach applied to problem (28).

1: Precondition: $\widetilde{x} \in \mathrm{conv}(X)$, $\widetilde{z} \in \mathrm{argmin}_z \{F(\widetilde{x}, z) : z \in Z\}$, $D \subseteq \mathrm{conv}(X)$
2: **function** SDM-GS(F, X, Z, D, $\widetilde{x}$, $\widetilde{z}$, t_{max})
3: **for** $t = 1, \ldots, t_{max}$ **do**
4: $\widetilde{x} \leftarrow \mathrm{argmin}_x \{F(x, \widetilde{z}) : x \in D\}$
5: $\widetilde{z} \leftarrow \mathrm{argmin}_z \{F(\widetilde{x}, z) : z \in Z\}$
6: **end for**
7: $\widehat{x} \in \mathrm{argmin}_x \{\nabla_x F(\widetilde{x}, \widetilde{z})^\top (x - \widetilde{x}) : x \in X\}$
8: Reconstruct D to be any set such that
9: $\{\widetilde{x} + \alpha(\widehat{x} - \widetilde{x}) : \alpha \in [0,1]\} \subseteq D \subseteq \mathrm{conv}(X)$
10: Set $\Gamma \leftarrow -\nabla_x F(\widetilde{x}, \widetilde{z})(\widehat{x} - \widetilde{x})$
11: **return** $(\widetilde{x}, \widetilde{z}, D, \Gamma)$
12: **end function**

If the z block update of Line 5 is trivialized, such as by making z not actually appear in the definition of F, or by making Z a singleton set, then Algorithm 2 would be identical to SDM applied to problem (28) in which the z block of variables correspondingly does not play any role. On the other hand, if the x update (4) is replaced with an update based on an exact minimization $\widetilde{x} \leftarrow \mathrm{argmin}_x \{F(x, \widetilde{z}) : x \in \mathrm{conv}(X)\}$ (so that the computations of Lines 7–10 and the returning of D and Γ can be skipped), then Algorithm 2 would be equivalent to a more traditional two-block nonlinear Gauss-Seidel method. Different forms of approximation of the x update, such as those resulting from gradient descent steps in x, are also considered in [30,11].

Remark 3 The main approach envisioned for constructing the inner approximation D on Lines 8–9 is to take $D \leftarrow \text{conv}(D \cup \{\widetilde{x}, \widehat{x}\})$. To implement this update of D, we need to save the points $\widehat{x}$ computed during previous calls to Algorithm 2.

We assume in the following proposition that Algorithm 2 is applied iteratively in the sense that at iteration $k \geqslant 0$, we input $(\widetilde{x}, \widetilde{z}) = (x^k, z^k)$ and return $(\widetilde{x}, \widetilde{z}) = (x^{k+1}, z^{k+1})$. Furthermore, at the same iteration k call of Algorithm 2, we set $d^{k+1} = \widehat{x} - \widetilde{x}$ where $\widehat{x}$ and $\widetilde{x}$ are set as in Line 9. This provides a reference sequence of directions $\{d^k\}$ necessary in the proof of the following proposition.

Proposition 3 *For problem* (28)*, let F be convex and continuously differentiable, and let* $\text{conv}(X)$ *and Z be nonempty and convex, with* $\text{conv}(X)$ *bounded and $z \mapsto F(x, z)$ inf-compact for each $x \in \text{conv}(X)$. Then, for any $t_{max} \geqslant 1$, the sequence $\{(x^k, z^k)\}$ generated by iterations of Algorithm 2 has limit points $(\bar{x}, \bar{z})$, each of which are optimal for problem* (28)*.*

Proof In light of the convexity and continuous differentiablity of F and the convexity of $\text{conv}(X)$ and Z, it is sufficient to show that

$$\nabla_x F(\bar{x}, \bar{z})^\top (x - \bar{x}) \geqslant 0 \quad \text{for all } x \in \text{conv}(X) \tag{29}$$

$$\text{and} \quad \nabla_z F(\bar{x}, \bar{z})^\top (z - \bar{z}) \geqslant 0 \quad \text{for all } z \in Z. \tag{30}$$

As $\nabla_z F(x^k, z^k)^\top (z - z^k) \geqslant 0$ for all $z \in Z$ holds for each $k \geqslant 1$ (this follows due to the optimality $z^k \in \text{argmin}_z \{F(x^k, z) : z \in Z\}$ that holds by construction) the satisfaction of the latter condition (30) is trivially established for any limit points $(\bar{x}, \bar{z})$. It remains only to show the satisfaction of the x-stationarity condition (29). This may be established by using Proposition 3.2 of [11] combined with the last sentence of Remark 3.3 from the same reference. But for the sake of explicitness, we use developments in Appendix A to show that (29) holds.

Note, for the sake of nontriviality, that $\nabla_x F(x^k, z^k)^\top (x - x^k) \geqslant 0$ for all $x \in X$ is assumed *not* to hold for any $k \geqslant 1$. Thus, with reference to the argument given in Appendix A, the sequence of directions $\{d^k\}$ satisfy the the Direction Assumption (DA), prior to Algorithm 4. Also, the Gradient Related Assumption (GRA) referred to in Appendix A is satisfied for this same $\{d^k\}$, by Lemma 7 therein. Due to the construction of D in Line 9 and setting $(x^{k+1}, z^{k+1}) = (\widetilde{x}, \widetilde{z})$ after the termination of the for loop of Lines 3–6, we have given $\{d^k\}$ and any choice of $(\beta, \sigma) \in (0, 1)$ the satisfaction of the Sufficient Decrease Assumption (SDA), also referred to in Appendix A. It then follows from Lemma 6 that limit points $(\bar{x}, \bar{z})$ of $\{(x^k, z^k)\}$ do exists, and that each of which satisfies the stationarity condition (29).

The method SDM-GS-ALM is now stated as Algorithm 3, which uses Algorithm 2 as a subroutine to provide a practical implementation of Algorithm 1.

Remark 4 At the return of Algorithm 2 in Line 8 of Algorithm 3, we have

$$\begin{aligned}\Gamma &= -\nabla_x L_\rho(x^k, z^k, \omega^k)^\top (\widehat{x} - x^k) \\ &= -\left[\nabla_x f(x^k) + \left(\omega^k + \rho(Qx^k - z^k)\right)^\top Q\right]^\top (\widehat{x} - x^k)\end{aligned}$$

where $\widehat{x}$ is computed on Line 7 of Algorithm 2. One may verify the equality $(Qx^k - z^k)^\top z^k = 0$ due to $z^k \in \operatorname{argmin}_z \left\{\|Qx^k - z\|_2^2 : z \in Z\right\}$. Moreover using this value of Γ and the computation of $\widetilde{\phi}$ on Lines 4 and 12 one may show, using the fact that $\hat{x} \in \operatorname{argmin}_x\{\nabla_x L_\rho(x^k, z^k, \omega^k)^\top (x - x^k) : x \in X\}$, that

$$\widetilde{\phi} = L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\|Qx^k - z^k\|_2^2 - \Gamma = \breve{\phi}\left(\omega^k + \rho(Qx^k - z^k), x^k\right).$$

Algorithm 3 A practical implementation of Algorithm 1 based on the use of SDM-GS iterations. (SDM-GS is given as Algorithm 2.)

1: **Preconditions:** $x^0 \in \operatorname{conv}(X)$, $z^0 \in Z$, $\omega^0 \in Z^\perp$, $D \subseteq \operatorname{conv}(X)$, $\gamma \in (0,1)$.
2: **function** SDM-GS-ALM(f, Q, X, Z, D, ρ, x^0, z^0, ω^0, γ, ϵ, t_{max}, k_{max})
3: $(x^0, z^0, D, \Gamma) \leftarrow$ SDM-GS($L_\rho(\cdot,\cdot,\omega^0)$, X, Z, D, x^0, z^0, t_{max})
4: $\widetilde{\phi} \leftarrow L_\rho(x^0, z^0, \omega^0) + \frac{\rho}{2}\|Qx^0 - z^0\|_2^2 - \Gamma$
5: set $\omega^0 \leftarrow \omega^0 + \rho(Qx^0 - z^0)$, $\breve{\phi}^0 \leftarrow \widetilde{\phi}$
6: **for** $k = 1, 2, \ldots, k_{max}$ **do**
7: Initialize $\omega^k \leftarrow \omega^{k-1}$, $\breve{\phi}^k \leftarrow \breve{\phi}^{k-1}$ ▷ (Default, null-step updates)
8: $(x^k, z^k, D, \Gamma) \leftarrow$ SDM-GS($L_\rho(\cdot,\cdot,\omega^k)$, X, Z, D, x^{k-1}, z^{k-1}, t_{max})
9: **if** $L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\|Qx^k - z^k\|_2^2 - \breve{\phi}^k \leqslant \epsilon$ **then**
10: **return** $(x^k, z^k, \omega^k, \breve{\phi}^k)$
11: **end if**
12: $\widetilde{\phi} \leftarrow L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\|Qx^k - z^k\|_2^2 - \Gamma$
13: $\eta_k \leftarrow \frac{\widetilde{\phi} - \breve{\phi}^k}{L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\|Qx^k - z^k\|_2^2 - \breve{\phi}^k}$
14: **if** $\eta_k \geqslant \gamma$ **then**
15: set $\omega^k \leftarrow \omega^k + \rho(Qx^k - z^k)$, $\breve{\phi}^k \leftarrow \widetilde{\phi}$
16: **end if**
17: Possibly update ρ, e.g., $\rho \leftarrow \frac{1}{\min\left\{\max\left\{(2/\rho)(1-\eta_k), 1/(10\rho), 10^{-4}\right\}, 10/\rho\right\}}$ as in [38]
18: **end for**
19: **return** $(x^k, z^k, \omega^k, \breve{\phi}^k)$
20: **end function**

Proposition 4 *Let $\left\{(x^k, z^k, \omega^k)\right\}$ be a sequence generated by Algorithm 3 applied to problem (1) with X compact, Z a linear subspace, $\omega^0 \in Z^\perp$, $\rho > 0$, $\gamma \in (0,1)$, $\epsilon = 0$ and $k_{max} = \infty$. If there exists a dual optimal solution ω^* to the dual problem (3), then either*

1. *$\omega^k = \overline{\omega}$ is fixed and optimal for (3) for $k \geqslant \bar{k}$ for some finite $\bar{k}$; or*
2. *ω^k is never optimal for (3) for any finite $k \geqslant 1$, but $\lim_{k\to\infty} \omega^k = \overline{\omega}$ is optimal,*

and the sequence $\left\{(x^k, z^k)\right\}$ has limit points $(\overline{x}, \overline{z})$, each of which is optimal for problem (2).

Proof In the first case, Algorithm 3 never takes serious steps for iterations $k \geqslant \bar{k} \geqslant 1$, and so with $\omega^k = \overline{\omega}$, optimal for (3) and fixed for $k \geqslant \bar{k}$, the Algorithm 3 iterations continue with the generation of $\left\{(x^k, z^k)\right\}$ as generated by iterations of SDM-GS (Algorithm 2). By Proposition 3, the sequence $\left\{(x^k, z^k)\right\}$ has limit points $(\overline{x}, \overline{z})$, each of which is optimal for problem (9) with $\omega = \overline{\omega}$. Then, by Lemma 1, $(\overline{x}, \overline{z})$ is also optimal for problem (2) since $\overline{\omega}$ is optimal for (3).

In the second case, where ω^k is never dual optimal for (2) for any finite $k \geqslant 1$, any serious step must be followed by a finite number of consecutive null-steps. We consider the subsequence indices $\{k_i\}_{i=1}^{\infty}$ where the update ω^{k_i+1} is obtained by a serious step. By Proposition 2, we have $\lim_{i\to\infty} \phi^C(\omega^{k_i+1}) = \zeta^{CLD}$, and accommodating the null steps in between, we have also $\lim_k \phi^C(\omega^k) = \zeta^{CLD}$. To prove the last claim, we note that $\omega^j = \omega^{k_{i+1}}$ for all integers j such that $k_i < j \leqslant k_{i+1}$ due to the taking of null steps. From Proposition 2, we have that $\lim_{i\to\infty} L_\rho(x^{k_i}, z^{k_i}, \omega^{k_i}) = \zeta^{CLD}$. By the continuity of $(x, z, \omega) \mapsto L_\rho(x, z, \omega)$, the convergences $\lim_{k\to\infty} \omega^k = \overline{\omega}$ and $\lim_{i\to\infty} Qx^{k_i} - z^{k_i} = 0$ (again, Proposition 2), we have $\lim_{i\to\infty} L_\rho(x^{k_i}, z^{k_i}, \omega^{k_{i+1}}) = \zeta^{CLD}$ also. Next, at each i, and integers j such that $k_i < j \leqslant k_{i+1}$, observe that

$$L_\rho(x^{k_i}, z^{k_i}, \omega^{k_{i+1}}) \geqslant L_\rho(x^j, z^j, \omega^j) \geqslant L_\rho(x^{k_{i+1}}, z^{k_{i+1}}, \omega^{k_{i+1}}).$$

In taking the limit of the above inequality as $i \to \infty$, it becomes evident that $\lim_{k\to\infty} L_\rho(x^k, z^k, \omega^k) = \zeta^{CLD}$ in the original sequence also. By the optimality of $\overline{\omega}$ for problem (3), we know from Lemma 1 that $\zeta_\rho^{AL}(\overline{\omega}) = \zeta^{CLD}$, and so each limit point $(\overline{x}, \overline{z})$ must be optimal for problem (9) with $\omega = \overline{\omega}$. Furthermore, by Lemma 1, $(\overline{x}, \overline{z})$ must also be optimal for problem (2). (These limit points exist furthermore, due to the compactness of $\mathrm{conv}(X)$ and the continuous and closed-form expression that the unique solution $z^k \in \mathrm{argmin}_z \left\{\left\|Qx^k - z\right\|_2^2 : z \in Z\right\}$ has given $x^k \in \mathrm{conv}(X)$ when Z is a linear subspace.)

4.1 Parallelization and workload

The opportunities for parallelization and distribution of the computational workload in SDM-GS-ALM, as stated in Algorithm 3, are not immediately apparent. This subsection explicitly indicates which update problems may be solved in parallel, and the nature of the required communication between the parallel computational nodes.

The bulk of computational work, parallelization, and parallel communication occurs within the SDM-GS method stated in Algorithm 2, where for the problems of interest, the following decomposable structures apply: $X = \prod_{i=1}^m X_i$, $D = \prod_{i=1}^m D_i$, and $F(x, z) = \sum_{i=1}^m F(x_i, z)$. In the larger context of Algorithm 3, the subproblem of Line 4 in Algorithm 2 can be solved in parallel given fixed $\widetilde{z} \in Z$ and $\omega \in Z^\perp$ along the block indices $i = 1, \ldots, m$ as

$$\min_x \left\{ f_i(x) + \omega_i^\top Q_i x + \frac{\rho}{2} \left\|Q_i x - \widetilde{z}_i\right\|_2^2 : x \in D_i \right\}, \tag{31}$$

while the subproblem of Line 7 is solved as

$$\min_x \left\{ \nabla_x f_i(\widetilde{x}_i) + (\omega_i + \rho(Q_i\widetilde{x}_i - \widetilde{z}_i))^\top Q_i x : x \in X_i \right\}.$$

Remark 5 In the setting where problem (1) is a large-scale mixed-integer linear optimization problem, the subproblems of Line 4 are continuous convex quadratic optimization problems for each block $i = 1, \ldots, m$, which can be solved independently of one another and in parallel. In the same setting, the Line 7 subproblems are mixed-integer optimization problems for each block $i = 1, \ldots, m$, which can also be solved independently of one another and in parallel. Additionally, the reconstruction of D occurring in Line 9 can be done in parallel for each D_i along the indices $i = 1, \ldots, m$.

Parallel communication is needed for the computation of the z update in Line 5 in Algorithm 2. In the larger context of Algorithm 3, this takes the form of solving

$$\min_z \left\{ \sum_{i=1}^{m} \|Q_i\widetilde{x}_i - z_i\|_2^2 : z \in Z \right\}.$$

This is solved as an averaging that requires the reduce-sum type parallel communication. The evaluation of the serious step condition through calculating η_k in Line 13 in Algorithm 3 also requires a reduce-sum type parallel communication. For implementation purposes, the computation of these values, including the computation of Γ from the SDM-GS call, can be combined into one reduce-sum communication. In total, each iteration of Algorithm 3 requires two reduce-sum type communications, one for computing the z-update of Line 5 Algorithm 2, and one combined reduce-sum communication to compute scalars associated with the Lagrangian bounds and the critical values for the termination conditions. The storage and updates of x^k and ω^k and D can also be done in parallel, while z^k and η_k need to be computed and stored by every processor at each iteration k.

5 Computational experiments and results

In this section, we present and examine the results of two computational tests with the following purposes:

Test 1: to demonstrate the effect of enforcing the serious step condition on the Lagrangian values;

Test 2: to compare the parallel speedup obtained with the use of two parallel implementations of SDM-GS-ALM (Algorithm 3) versus the parallel speedups reported in [41] for two other parallel approaches. Additionally, the final iteration Lagrangian bounds are compared between the different parallel implementations for each experiment.

Computational experiments were performed on instances from three classes of problems. The first class consists of the capacitated allocation problems (CAP) [9]. The second and third classes consist of the DCAP and SSLP problems from the Stochastic Integer Programming Test Problem Library (SIPLIB), which are described in detail in [47,2] and accessible at [2]. These are all large-scale mixed-integer linear optimization problems, so the preceding observations for when f is linear apply.

Test 1 was conducted with a Matlab 2012b [44] serial implementation of Algorithm 3 using CPLEX 12.6.1 [35] as the solver. The computing environment was on an Intel® Core™ i7-4770 3.40 GHz processor with 8 GB RAM and on a 64-bit operating system. All experiments for Test 1 were run with maximum number of iterations $k_{max} = 20$.

The parallel experiments of Test 2 were conducted with a C++ implementation of Algorithm 3 using CPLEX 12.5 [36] as the solver and the message passing interface (MPI) for parallel communication. For reading SMPS files into scenario-specific subproblems and for their interface with CPLEX, we used modified versions of the COIN-OR [1] Smi and Osi libraries, either to instantiate appropriate C++ class instances of the subproblems directly, or to write scenario-specific MPS files from the SMPS file. The computing environment for the Test 2 experiments is the Raijin cluster maintained by Australia's National Computing Infrastructure (NCI) and supported by the Australian government [46]. The Raijin cluster is a high performance computing (HPC) environment which has 3592 nodes (system units), 57472 cores of Intel Xeon E5-2670 processors with up to 8 GB PC1600 memory per core (128 GB per node). All experiments were conducted using one thread per CPLEX solve.

5.1 Effects of the Serious Step Condition

The results of the Test 1 set of experiments are depicted in the plots of Figure 1 (with additional Figures 2, 3 and 4 in Appendix B). The use of different penalty parameter ρ values is differentiated by the use of different plot colors. The penalties are chosen so that the smallest penalties (in red) are near optimal in terms of the resulting computational performance, while the larger penalties are known beforehand to be too large for optimal performance. For testing purposes, this is the most interesting way to choose penalty values, as smaller (than optimal) penalty values yield very little difference in Lagrangian bound between the use of different SSC parameter values. Solid line and dashed line plots depict the Lagrange bounds due to the use of a more stringent SSC parameter value $\gamma = 0.5$ and a more lenient value for the SSC parameter $\gamma = 0.125$, respectively. The dotted line plots depict the Lagrangian values resulting from the non-use of the SSC, so that it evaluates true no matter what. The following observations are suggested from the results of the Test 1 experiments:

1. First, the most significant differences between the varied use of SSC occur when the penalty coefficient values are large. In this setting, it seems to

be the case that the use of more stringent (i.e., larger) values of the SSC parameter γ has the effect of mitigating the destabilizing effect of having a penalty parameter ρ value that is too large. This is significant because the performance of iterative Lagrangian dual solution approaches based on (or related to) proximal bundle methods is sensitive to the tuning of the ρ value, and the optimal tuning of such parameters is assumed to be unknown beforehand in practical applications. For this reason, any mechanism to mitigate the effect of having an unfavorable tuning of the penalty parameter is highly desirable.

2. As is the case for the proximal bundle method, information from the SSC test can be used to dynamically fine-tune the value of the penalty parameter ρ.
3. While not enforcing the SSC can adversely affect the growth trend in the Lagrangian bound, the use of a SSC parameter γ value that is too large can have a similar effect for the tail-end values. This is most clearly seen in the Figure 1 DCAP-233-500 $\rho = 50$ and $\rho = 100$ plots. In these plots, the growth in Lagrangian bound value is noticeably stunted in the tail-end iterations for the larger $\gamma = 0.5$ value as compared with the smaller $\gamma = 0.125$.

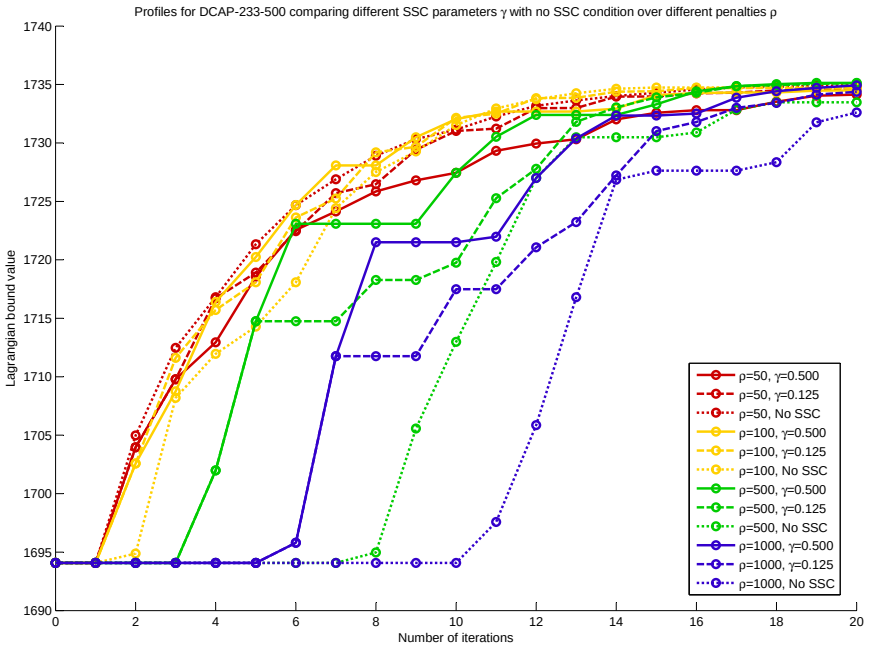

Fig. 1 Applying SDM-GS-ALM using different parameterizations for the SSC condition (or none).

5.2 Benefits of Parallelization

For the Test 2 experiments, we primarily compare the parallel speedup achieved with Algorithm 3 against that achieved with the enhancements to the proximal bundle method presented in [41]. Additionally, we compare the Lagrangian bound at the final iteration.

The enhancements in [41] use structure-exploiting primal-dual interior point solvers to improve the parallel efficiency of solving the proximal bundle method master problem. (The solution of this master problem is analogous to the approximated solution to problem (9) obtained by using the SDM-GS method in Algorithm 2.) The first solver is referred to by its acronym OOQP [25], while the second is PIPS-IPM [42].

In the experiments of Test 2, the underlying computing architecture and third-party software are inevitably different between our tests and those in [41]. Additionally, the termination criterion is necessarily different from that given in Step 2 of Figure 2 in [41] due to the differences in algorithms. In our tests, the termination criterion comes from Lines 9–11 of Algorithm 3 with $\epsilon = 10^{-6}$. We can nevertheless create a meaningful control in the tuning of the most important parameters affecting the performance of the algorithm.

1. As done in [41], we set the SSC parameter $\gamma = 0.1$, and we initialize the dual solution $\omega^0 = 0$.
2. In analogy to the possible trimming of cutting planes noted in [41], practical implementations of Algorithm 3 may judiciously trim the set D to improve performance. As all cuts are kept in the experiments of [41], so we also avoid trimming the expansion of D in our experiments, and so we just use the simple update rule $D \leftarrow \text{conv}(D \cup \{\widetilde{x}, \widehat{x}\})$ within Algorithm 2.
3. We use an update rule analogous to the one in [38] as is done in [41]. which takes the suggested form given in Line 17 of Algorithm 3. Initially, $\rho = 1$.

In Tables 1–2, the columns headed by OOQP and PIPS-IPM report the parallel speedup due to the use of $N = 1, 8, 16, 32$ processors, which are originally reported in Figure 2 of [41]. If, given the use of N processors, T_N denotes the total wall clock time (in seconds) divided by number of iterations, then we compute the parallel speedup as T_1/T_N. For the computational experiments with Algorithm 3, we compute each table entry T_1/T_N after taking, from five identically parameterized experiments, 1) the minimum T_1 value, and 2) the average T_N, $N > 1$, value. The column headed by SDM-GS1-ALM presents the parallel speedup values for the application of Algorithm 3 with $t_{max} = 1$. The column headed by SDM-GS5-ALM is analogous, with $t_{max} = 5$. The total wall clock time per iteration values used to compute the ratios T_1/T_N are provided in Appendix C, accounting for taking the minimum ($N = 1$) or average ($N > 1$) over the five experiments for each set of parameterizations associated with Algorithm 3. For the two sets of experiments based on the application of Algorithm 3, a problem-specific maximum number of main loop iterations was set so as to make the tests as comparable with the tests in [41] as possible. These data are also reported in Appendix C. Also in Tables 1–2, the best La-

grangian bounds obtained for each combination of test problem and algorithm are reported.

	Speedup for SSLP 5-25-100			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	5.54	5.23	4.38	4.78
16	8.89	8.55	6.61	7.07
32	11.69	11.94	8.19	8.89
Lagr. Value	-127.37	-127.37	-127.71	-127.58
	Speedup for SSLP 10-50-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.64	2.80	6.87	6.95
16	2.70	2.92	12.95	12.84
32	2.98	3.40	21.67	20.98
Lagr. Value	-349.14	-349.14	-349.48	-349.14

	Speedup for SSLP 10-50-2000	
No. Proc.	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00
2	2.34	2.34
4	4.81	4.83
8	9.29	9.25
16	18.69	18.48
32	34.63	35.10
64	60.59	60.93
Lagr. Value	-348.35	-347.75

Table 1 SSLP: Comparing speedup and final best Lagrangian bound

We draw the following conclusions from the results of the Test 2 experiments reported in Tables 1–2.

1. The improvement in parallel speedup (SDM-GS-ALM columns) over either OOQP or PIPS-IPM is evident for all problems except for the one with the fewest number of scenarios (SSLP 5-25-100).
2. Slightly inferior final Lagrange bounds reported for SDM-GS1-ALM ($t_{max} = 1$) are evident. This deficit is improved by using SDM-GS with $t_{max} = 5$, as done for the SDM-GS5-ALM experiments. But even these bounds are usually not as good as the bounds obtained with OOQP or PIPS-IPM; this is due to their more exact solving of the master problem instances. This suggests that as the iterations $k \geqslant 1$ increase, it is advantageous to solve the continuous master problem with SDM-GS iterations using larger t_{max} values.
3. Interestingly, parallel speedup is enhanced for SDM-GS5-ALM over SDM-GS1-ALM; although the latter yields lower average total wall clock time per iteration, the proportion of efficiently parallelizable work seems to increase in the former.

For Test 2, we also tested the performance of Algorithm 3 on the SSLP 10-50-2000 problem, which is of substantially larger scale than the other test

	Speedup for DCAP 233-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.44	5.32	6.88	8.11
16	2.81	8.15	13.28	15.65
32	1.63	10.25	23.42	27.40
Lagr. Value	1736.68	1736.68	1734.99	1736.02
	Speedup for DCAP 243-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.85	5.71	6.51	7.61
16	3.59	5.85	12.28	14.44
32	1.98	6.44	21.99	25.25
Lagr. Value	2165.48	2165.50	2162.58	2164.48
	Speedup for DCAP 332-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.03	5.56	6.83	8.50
16	2.33	5.00	12.84	16.20
32	1.21	6.61	21.83	23.48
Lagr. Value	1587.44	1587.44	1584.77	1586.11
	Speedup for DCAP 342-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.45	3.78	7.16	8.25
16	2.71	4.36	12.95	15.49
32	1.84	4.64	22.41	26.93
Lagr. Value	1902.84	1903.21	1900.81	1901.90

Table 2 DCAP: Comparing speedup and final best Lagrangian bound

problems considered in this paper. Using $N = 1, 2, 4, 8, 16, 32, 64$ processors, we see very good speedup, which suggests the realized benefit of distributing the use of memory. We also see that for such large-scale problems, the additional cost in time of performing more inner loop Gauss-Seidel iterations (larger t_{max}) becomes marginal, since the cost of solving the mixed-integer linear subproblems takes a larger share of the computational time.

6 Conclusion and future work

Our contribution is motivated by the goal of improving the efficiency of parallelization applied to iterative approaches for solving the Lagrangian dual problem of large scale optimization problems. These problems have nonlinear convex differentiable objective f, decomposable nonconvex constraint set X, and nondecomposable affine constraint set $Qx = z$ to which Lagrangian relaxation is applied. Problems of such a form include the split variable extensive form of mixed-integer linear stochastic programs as a special case. Implicitly, our approach refers to the convex hull $\text{conv}(X)$ of X, and the assumed lack of known description of $\text{conv}(X)$ needs to be addressed. Proximal bundle methods (alternatively in the form of the proximal simplicial decomposition method

or stabilized column generation) are well-known for addressing the latter issue. In the former issue, that of exploiting the large scale structure to apply parallel computation efficiently, we develop a modified augmented Lagrangian (AL) method with approximate subproblem solutions that incorporates ideas from the proximal bundle method.

The approximation of subproblem solutions is based on an iterative approach that integrates ideas from the simplicial decomposition method (SDM) (for constructing inner approximations of $\text{conv}(X)$) and the nonlinear block Gauss-Seidel method. It is the latter Gauss-Seidel aspect that is primarily responsible for enhancing the parallel efficiency that is observed in the numerical experiments. While convergence analysis of the integrated SDM-GS approach may be derived from slight modifications to results in [11], for the sake of completeness and explicitness, we provide in the appendix a proof of optimal convergence of SDM-GS as it is applied within our algorithm under a standard set of conditions. A distinction between so-called "serious" steps and "null" steps, in analogy to the proximal bundle method, is also recovered. Once these aspects are successfully integrated, then the contribution is complete, where the beneficial stabilization associated with proximal point methods and the ability to apply parallelization more efficiently are both realized. The resulting algorithm developed in this paper is referred to as SDM-GS-ALM, which has similar functionality to the alternating direction method of multipliers (ADMM).

We performed numerical tests of two sorts. In Test 1, we examined the impact of varying the serious step condition parameter. We found that parameterizations that effect more stringent serious step conditions seem to have the effect of mitigating the early iteration instability due to penalty parameters that are too large. At the same time, the more stringent serious step condition parameterizations seemed to result in slower convergence to dual optimality in the tail-end. As is the case for proximal bundle methods, information obtained in the serious step condition tests may be used to beneficially adjust the proximal term penalty coefficient in early iterations.

In Test 2, we examined the efficiency of parallelization, measured by the speedup ratio, due to the use of the SDM-GS-ALM, compared versus preexisting implementations of the proximal bundle method that use structure exploiting primal dual interior point methods to improve parallel efficiency. We saw in these results a promising increase in parallel efficiency due to the use of SDM-GS-ALM, where the increase in parallel efficiency is attributed primarily to the successful incorporation of Gauss-Seidel iterations. The results of the last problem tested, SSLP 10-50-2000, additionally suggested a benefit due to the ability of SDM-GS-ALM to distribute not just the workload, but also the use of memory. The vector of auxiliary variables z is the only substantial block of data that needs to be stored and modified by all processors. In the context of stochastic optimization problems, this represents a modest communication bottleneck in proportion to the number of first-stage variables for two-stage problems, while for multistage problems, the amount of such data that must be

stored by every processor and modified by parallel communication can increase exponentially with the number of stages.

Potential future improvements include the following. While a default implementation of SDM-GS-ALM would have one Gauss-Seidel iteration per SDM-GS call, the Lagrangian bounds reported from the Test 2 experiments suggest that an improved implementation would have early iterations use one Gauss-Seidel iteration per SDM-GS call, but steadily increase the number of Gauss-Seidel iterations per SDM-GS call for the later iterations. This results in better Lagrangian bounds at termination. While these extra Gauss-Seidel iterations require extra parallel communication, the additional wall clock time required becomes increasingly marginal for larger problems where the cost of solving the SDM linearized subproblems associated with expanding the inner approximation increasingly outweighs the cost associated with computing the approximate solution of the continuous master problem and any required parallel communications.

A potentially large improvement to the speed of convergence, in terms of wall clock time, would be to incorporate into the analysis the degree to which the SDM linearized subproblem can be solved suboptimally and yet retain the optimal convergence. We expect that solving these subproblems exactly, particularly in the early iterations, is highly wasteful, and providing a theoretical basis for controlling the tolerance of solution inaccuracy would be of great value. Another potential avenue for future work is to extend the experimental analysis to multistage mixed-integer stochastic optimization problems and/or nonlinear problems, as the form of the problem addressed by SDM-GS-ALM is general enough to model these types of problems.

A Technical lemmas for establishing optimal convergence of SDM-GS

Given initial $(x^0, z^0) \in X \times Z \subset \mathbb{R}^n \times \mathbb{R}^q$, we consider the generation of the sequence $\{(x^k, z^k)\}$ with iterations computed using Algorithm 4, whose target problem is given by

$$\min_{x,z} \{F(x,z) : x \in X, z \in Z\}, \tag{32}$$

where $(x,z) \mapsto F(x,z)$ is convex and continuously differentiable over $X \times Z$, and sets X and Z are closed and convex, with X bounded and $z \mapsto F(x,z)$ is inf-compact for each $x \in X$.

We define the directional derivative with respect to x as

$$F'_x(x,z;d) := \lim_{\alpha \downarrow 0} \frac{F(x+\alpha d, z) - F(x,z)}{\alpha}.$$

Of interest is the satisfaction of the following local stationarity condition at $x \in X$:

$$F'_x(x,z;d) \geqslant 0 \quad \text{for all } d \in X - \{x\} \tag{33}$$

for any limit point $(x,z) = (\bar{x}, \bar{z})$ of some sequence $\{(x^k, z^k)\}$ of feasible solutions to problem (32). For the sake of nontriviality, we shall assume that the x-stationarity condition (33) never holds at $(x,z) = (x^k, z^k)$ for any $k \geqslant 0$. Thus, for each x^k, $k \geqslant 0$, there always exists a $d^k \in X - \{x^k\}$ for which $F'_x(x^k, z^k; d^k) < 0$.

Direction Assumptions (DAs): For each iteration $k \geqslant 0$, given $x^k \in X$ and $z^k \in Z$, we have d^k chosen so that 1) $x^k + d^k \in X$; and 2) $F'_x(x^k, z^k; d^k) < 0$.

Gradient Related Assumption (GRA): Given a sequence $\{(x^k, z^k)\}$ with $\lim_{k\to\infty}(x^k, z^k) = (\overline{x}, \overline{z})$, and a bounded sequence $\{d^k\}$ of directions, then the existence of a direction $\overline{d} \in X - \{\overline{x}\}$ such that $F'_x(\overline{x}, \overline{z}; \overline{d}) < 0$ implies that

$$\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0. \tag{34}$$

In this case, we say that $\{d^k\}$ is *gradient related* to $\{x^k\}$. This gradient related condition is similar to the one defined in [5]. The sequence of directions d^k is typically gradient related to $\{x^k\}$ by construction. (See Lemma 7.)

To state the last assumption, we require the notion of an Armijo rule step length $\alpha^k \in (0, 1]$ given (x^k, z^k, d^k) and parameters $\beta, \sigma \in (0, 1)$.

Algorithm 4 Computing an Armijo rule step length α^k at iteration k.

1: **function** ARMIJOSTEP(F, x^k, z^k, d^k, β, σ)
2: $\quad\alpha^k \leftarrow 1$
3: $\quad$**while** $F(x^k + \alpha^k d^k, z^k) - F(x^k, z^k) > \alpha^k \sigma F'_x(x^k, z^k; d^k)$ **do**
4: $\quad\quad\alpha^k \leftarrow \beta\alpha^k$
5: $\quad$**end while**
6: $\quad$**return** α^k
7: **end function**

Remark 6 Under mild assumptions on F such as continuity that guarantee the existence of finite $F'_x(x, z; d)$ for all $(x, z, d) \in \{(x, z, d) : x \in X, d \in X - \{x\}, z \in Z\}$, we may assume that the while loop of Lines 3–5 terminates after a finite number of iterations. Thus, we have $\alpha^k \in (0, 1]$ for each $k \geqslant 1$.

The last significant assumption is stated as follows.

Sufficient Decrease Assumption (SDA): For sequences $\{(x^k, z^k, d^k)\}$ and step lengths $\{\alpha^k\}$ computed according to Algorithm 4, we assume for each $k \geqslant 0$, that (x^{k+1}, z^{k+1}) satisfies

$$F(x^{k+1}, z^{k+1}) \leqslant F(x^k + \alpha^k d^k, z^k).$$

Lemma 6 *For problem* (32)*, let $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \mapsto \mathbb{R}$ be convex and continuously differentiable, $X \subset \mathbb{R}^{n_x}$ convex and compact, and $Z \subseteq \mathbb{R}^{n_z}$ closed and convex. Furthermore, assume for each $x \in X$ that $z \mapsto F(x, z)$ is inf-compact. If a sequence $\{(x^k, z^k, d^k)\}$ satisfies the DA, the GRA, and the SDA for some fixed $\beta, \sigma \in (0, 1)$, then the sequence (x^k, z^k) has limit points $(\overline{x}, \overline{z})$, each of which satisfies the stationarity condition* (33)*.*

Proof The existence of limit points $(\overline{x}, \overline{z})$ follows from the compactness of X, the inf-compactness of $z \mapsto F(x, z)$ for each $x \in X$, and the SDA. In generating $\{\alpha^k\}$ according to the Armijo rule as implemented in Lines 2–5 of Algorithm 4, we have

$$\frac{F(x^k + \alpha^k d^k, z^k) - F(x^k, z^k)}{\alpha^k} \leqslant \sigma F'_x(x^k, z^k; d^k). \tag{35}$$

By the DA, $F'_x(x^k, z^k; d^k) < 0$ and since $\alpha^k > 0$ for each $k \geqslant 1$ by Remark 6, we infer from (35) that $F(x^k + \alpha^k d^k, z^k) < F(x^k, z^k)$. By construction, we have $F(x^{k+1}, z^{k+1}) \leqslant F(x^k + \alpha^k d^k, z^k) < F(x^k, z^k)$.. By the monotonicity $F(x^{k+1}, z^{k+1}) < F(x^k, z^k)$ and F being bounded from below on $X \times Z$, we have $\lim_{k\to\infty} F(x^k, z^k) = \bar{F} > -\infty$. Therefore,

$$\lim_{k\to\infty} F(x^{k+1}, z^{k+1}) - F(x^k, z^k) = 0,$$

which implies

$$\lim_{k\to\infty} F(x^k + \alpha^k d^k, z^k) - F(x^k, z^k) = 0. \tag{36}$$

We assume for sake of contradiction that $\lim_{k\to\infty}(x^k, z^k) = (\overline{x}, \overline{y})$ does not satisfy the stationarity condition (33). By GRA, we have that $\{d^k\}$ is gradient related to $\{x^k\}$; that is,

$$\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0. \tag{37}$$

Thus, it follows from (35)–(37) that $\lim_{k\to\infty} \alpha^k = 0$.

Consequently, after a certain iteration $k \geqslant \bar{k}$, we can define $\{\bar{\alpha}^k\}$, $\bar{\alpha}^k = \alpha^k/\beta$, where $\bar{\alpha}^k \leqslant 1$ for $k \geqslant \bar{k}$, and so we have

$$\sigma F'_x(x^k, z^k; d^k) < \frac{F(x^k + \bar{\alpha}^k d^k, z^k) - F(x^k, z^k)}{\bar{\alpha}^k}. \tag{38}$$

Since F is continuously differentiable, the mean value theorem may be applied to the right-hand side of (38) to get

$$\sigma F'_x(x^k, z^k; d^k) < F'_x(x^k + \tilde{\alpha}^k d^k, z^k; d^k), \tag{39}$$

for some $\tilde{\alpha}^k \in [0, \overline{\alpha}^k]$.

Again, using the assumption $\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0$, and also the compactness of $X - X$, we take a limit point $\overline{d}$ of $\{d^k\}$, with its associated subsequence index set denoted by $\mathcal{K}$, such that $F'_x(\overline{x}, \overline{z}, \overline{d}) < 0$. Taking the limits over the subsequence indexed by $\mathcal{K}$, we have $\lim_{k\to\infty, k\in\mathcal{K}} F'_x(x^k, z^k; d^k) = F'_x(\overline{x}, \overline{z}; \overline{d})$ and $\lim_{k\to\infty, k\in\mathcal{K}} F'_x(x^k + \tilde{\alpha}^k d^k, z^k; d^k) = F'_x(\overline{x}, \overline{z}; \overline{d})$. These two limits holds since 1) $(x, z) \mapsto F'_x(x, z; d)$ for each $d \in X - X$ is continuous and 2) $d \mapsto F'_x(x, z; d)$ is locally Lipschitz continuous for each $(x, z) \in X \times Z$ (e.g., Proposition 2.1.1 of [16]); these two facts together imply that $(x, z; d) \mapsto F'_x(x, z; d)$ is continuous. Then, inequality (39) becomes in the limit as $k \to \infty$, $k \in \mathcal{K}$,

$$\sigma F'_x(\overline{x}, \overline{z}; \overline{d}) \leqslant F'_x(\overline{x}, \overline{z}; \overline{d}) \quad \Longrightarrow \quad 0 \leqslant (1 - \sigma) F'_x(\overline{x}, \overline{z}; \overline{d}).$$

Since $(1 - \sigma) > 0$ and $F'_x(\overline{x}, \overline{z}; \overline{d}) < 0$, we have a contradiction. Thus, $\overline{x}$ must satisfy the stationary condition (33).

Remark 7 Noting that $F'_x(x^k, z^k; d^k) = \nabla_x F(x^k, z^k)^\top d^k$ under the assumption of continuous differentiability of F, one means of constructing $\{d^k\}$ is as follows:

$$d^k \leftarrow \operatorname*{argmin}_d \left\{ \nabla_x F(x^k, z^k)^\top d : d \in X - \left\{x^k\right\} \right\}. \tag{40}$$

Lemma 7 *Given sequence $\{(x^k, z^k)\}$ with $\lim_{k\to\infty}(x^k, z^k) = (\overline{x}, \overline{z})$, let each d^k, $k \geqslant 1$, be generated as in* (40)*. Then $\{d^k\}$ is gradient related to $\{x^k\}$.*

Proof By the construction of d^k, $k \geqslant 1$, we have

$$F'_x(x^k, z^k; d^k) \leqslant F'_x(x^k, z^k; d) \quad \forall\, d \in X - \left\{x^k\right\}.$$

Taking the limit, we have

$$\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) \leqslant \limsup_{k\to\infty} F'_x(x^k, z^k; d) \leqslant F'_x(\overline{x}, \overline{z}; d) \quad \forall\, d \in X - \{\overline{x}\},$$

where the last inequality follows from the upper semicontinuity of the function $(x, z, d) \mapsto F'_x(x, z; d)$, which holds in our setting due, primarily, to Proposition 2.1.1 (b) of [16] given that F is assumed to be convex and continuous on $\mathbb{R}^n$. Taking

$$\overline{d} \in \operatorname*{argmin}_d \left\{ F'_x(\overline{x}, \overline{z}; d) : d \in X - \{\overline{x}\} \right\},$$

we have by the assumed nonstationarity that $F'_x(\overline{x}, \overline{z}; \overline{d}) < 0$. Thus, $\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0$, and so GRA holds.

References

1. COmputational INfrastructure for Operations Research. URL http://www.coin-or.org/. Last accessed 28 January, 2016
2. Ahmed, S., Garcia, R., Kong, N., Ntaimo, L., Parija, G., Qiu, F., Sen, S.: SIPLIB: A stochastic integer programming test problem library (2015). URL http://www.isye.gatech.edu/ sahmed/siplib
3. Amor, H.B., Desrosiers, J., Frangioni, A.: On the choice of explicit stabilizing terms in column generation. Discrete Applied Mathematics **157**(6), 1167 – 1184 (2009)
4. Bertsekas, D.: Constrained Optimization and Lagrange Multiplier Methods. Academic Press (1982)
5. Bertsekas, D.: Nonlinear Programming. Athena Scientific (1999)
6. Bertsekas, D.: Convex Optimization Algorithms. Athena Scientific (2015)
7. Bertsekas, D.: Incremental aggregated proximal and augmented Lagrangian algorithms. arXiv preprint arXiv:1509.09257 (2015)
8. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer Science & Business Media (2011)
9. Bodur, M., Dash, S., Günlük, O., Luedtke, J.: Strengthened Benders cuts for stochastic integer programs with continuous recourse (2014). URL http://www.optimization-online.org/DB_FILE/2014/03/4263.pdf. Last accessed on 13 January 2015
10. Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., Oliveira, F.: Progressive hedging with a Frank-Wolfe based method for computing stochastic mixed-integer programming Lagrangian dual bounds. Optimization Online (2016). URL http://www.optimization-online.org/DB_HTML/2016/03/5391.html
11. Bonettini, S.: Inexact block coordinate descent methods with application to non-negative matrix factorization. IMA Journal of Numerical Analysis **31**(4), 1431–1452 (2011)
12. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundation and Trends in Machine Learning **3**(1), 1–122 (2011)
13. Carøe, C.C., Schultz, R.: Dual decomposition in stochastic integer programming. Operations Research Letters **24**(1), 37–45 (1999)
14. Chatzipanagiotis, N., Dentcheva, D., Zavlanos, M.: An augmented Lagrangian method for distributed optimization. Mathematical Programming **152**(1), 405–434 (2014)
15. Chen, G., Teboulle, M.: A proximal-based decomposition method for convex minimization problems. Mathematical Programming **64**, 81–101 (1994)
16. Clarke, F.: Optimization and Nonsmooth Analysis. Society for Industrial and Applied Mathematics (1990)
17. Eckstein, J.: A practical general approximation criterion for methods of multipliers based on Bregman distances. Mathematical Programming **96**(1), 61–86
18. Eckstein, J., Bertsekas, D.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Mathematical Programming **55**(1-3), 293–318 (1992)
19. Eckstein, J., Silva, P.: A practical relative error criterion for augmented lagrangians. Mathematical Programming **141**(1), 319–348 (2013)
20. Eckstein, J., Yao, W.: Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. Tech. rep., Rutgers University (2014)
21. Feizollahi, M.J., Costley, M., Ahmed, S., Grijalva, S.: Large-scale decentralized unit commitment. International Journal of Electrical Power & Energy Systems **73**, 97–106 (2015)
22. Fischer, F., Helmberg, C.: A parallel bundle framework for asynchronous subspace optimization of nonsmooth convex functions. SIAM Journal on Optimization **24**(2), 795–822 (2014)
23. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Computers and Mathematics with Applications **2**, 17–40 (1976)

24. Gade, D., Hackebeil, G., Ryan, S.M., Watson, J.P., Wets, R.J.B., Woodruff, D.L.: Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. Mathematical Programming **157**(1), 47–67 (2016)
25. Gertz, E., Wright, S.: Object-oriented software for quadratic programming. ACM Transactions on Mathematical Software **29**(1), 58–81 (2003)
26. Glowinski, R., Marrocco, A.: Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problems de dirichlet non lineares. Revue Française d'Automatique, Informatique, et Recherche Opérationelle **9**, 41–76 (1975)
27. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Operations Research Letters **26**(3), 127–136 (2000)
28. Hamdi, A., Mahey, P., Dussault, J.P.: Recent Advances in Optimization: Proceedings of the 8th French-German Conference on Optimization Trier, July 21–26, 1996, chap. A New Decomposition Method in Nonconvex Programming via a Separable Augmented Lagrangian, pp. 90–104. Springer Berlin Heidelberg, Berlin, Heidelberg (1997)
29. Hare, W., Sagastizábal, C., Solodov, M.: A proximal bundle method for nonsmooth nonconvex functions with inexact information. Computational Optimization and Applications **63**, 1–28 (2016)
30. Hathaway, R.J., Bezdek, J.C.: Grouped coordinate minimization using Newton's method for inexact minimization in one vector coordinate. Journal of Optimization Theory and Applications **71**(3), 503–516 (1991)
31. He, B., Liao, L.Z., Han, D., Yang, H.: A new inexact alternating directions method for monotone variational inequalities. Mathematical Programming **92**, 103–118 (2002)
32. Hestenes, M.R.: Multiplier and gradient methods. Journal of Optimization Theory and Applications pp. 303–320 (1969)
33. Hildreth, C.: A quadratic programming procedure. Naval Research Logistics Quarterly **4**, 79–85, 361 (1957)
34. Holloway, C.: An extension of the Frank and Wolfe method of feasible directions. Mathematical Programming **6**(1), 14–27 (1974)
35. IBM Corporation: IBM ILOG CPLEX Optimization Studio CPLEX Users Manual. URL http://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.6.1/ilog.odms.studio.help/pdf/usrcplex.pdf. Last accessed 22 August 2016
36. IBM Corporation: IBM ILOG CPLEX V12.5. URL http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/. Last accessed 28 Jan 2016
37. Kiwiel, K., Rosa, C., Ruszczyński, A.: Proximal decomposition via alternating linearization. SIAM Journal on Optimization **9**(3), 668–689 (1999)
38. Kiwiel, K.C.: Approximations in proximal bundle methods and decomposition of convex programs. Journal of Optimization Theory and Applications **84**(3), 529–548 (1995)
39. Lemaréchal, C.: An extension of Davidon methods to non differentiable problems, pp. 95–109. Springer Berlin Heidelberg (1975)
40. Lin, X., Pham, M., Ruszczyński, A.: Alternating linearization for structured regularization problems. Journal of Machine Learning Research **15**, 3447–3481 (2014)
41. Lubin, M., Martin, K., Petra, C., Sandıkçı, B.: On parallelizing dual decomposition in stochastic integer programming. Operations Research Letters **41**(3), 252–258 (2013)
42. Lubin, M., Petra, C., Anitescu, M., Zavala, V.: Scalable stochastic optimization of complex energy systems. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, p. 64:164:10. ACM, Seattle, WA (2011)
43. Mahey, P., Oualibouch, S., Tao, P.D.: Proximal decomposition on the graph of a maximal monotone operator. SIAM Journal on Optimization **5**(2), 454–466 (1995)
44. The MathWorks, Natick: MATLAB 2012b (2014)
45. Mulvey, J., Ruszczyński, A.: A diagonal quadratic approximation method for large scale linear programs. Operations Research Letters **12**(4), 205 – 215 (1992)
46. National Computing Infrastructure (NCI): NCI Website. URL http://www.nci.org.au. Last accessed 19 November 2016
47. Ntaimo, L.: Decomposition algorithms for stochastic combinatorial optimization: Computational experiments and extensions. Ph.D. thesis (2004)

48. de Oliveira, W., Sagastizábal, C.: Bundle Methods in the XXIst Century: A Bird's-eye view. Pesquisa Operacional **34**, 647–670 (2014)

49. de Oliveira, W., Sagastizábal, C., Lemaréchal, C.: Convex proximal bundle methods in depth: a unified analysis for inexact oracles. Mathematical Programming **148**(1), 241–277 (2014)

50. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: R. Fletcher (ed.) Optimization. New York: Academic Press (1969)

51. Rockafellar, R.: Convex Analysis. Princeton University Press (1970)

52. Rockafellar, R.: Monotone operators and the proximal point algorithm. SIAM Journal on Control and Optimization **14**(5), 877–898 (1976)

53. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Mathematics of Operations Research **1**(2), 97–116 (1976)

54. Ruszczyński, A.: Nonlinear Optimization. Princeton University Press (2006)

55. Ruszczyński, A.: On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. Mathematics of Operations Research **20**(3), 634–656 (1995)

56. Shor, N.: Minimization Methods for Non-differentiable Functions. Springer-Verlag, New York (1985)

57. Tappenden, R., Richtárik, P., Büke, B.: Separable approximations and decomposition methods for the augmented Lagrangian. Optimization Methods & Software **30**(3), 643–668 (2015)

58. Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of Optimization Theory and Applications **109**, 475–494 (2001)

59. Von Hohenbalken, B.: Simplicial decomposition in nonlinear programming algorithms. Mathematical Programming **13**(1), 49–68 (1977)

60. Warga, J.: Minimizing certain convex functions. SIAM Journal on Applied Mathematics **11**, 588–593 (1963)

B Supplementary Material: Additional Figures

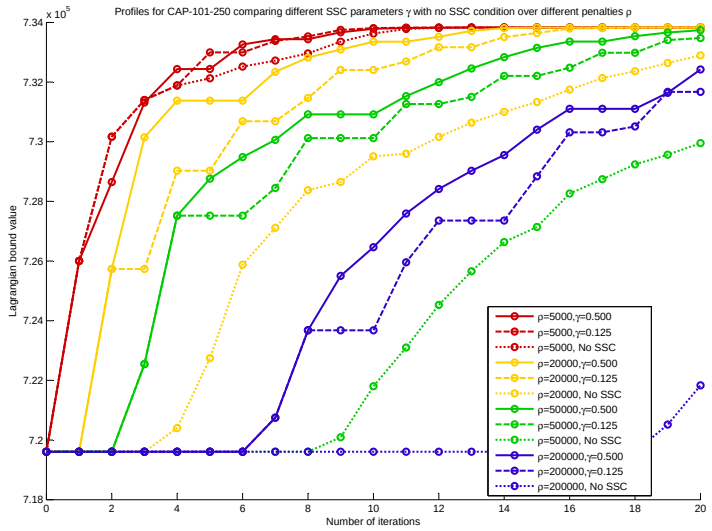

Fig. 2 Applying SDM-GS-ALM to CAP-101-250 using different parameterizations for the SSC condition (or none).

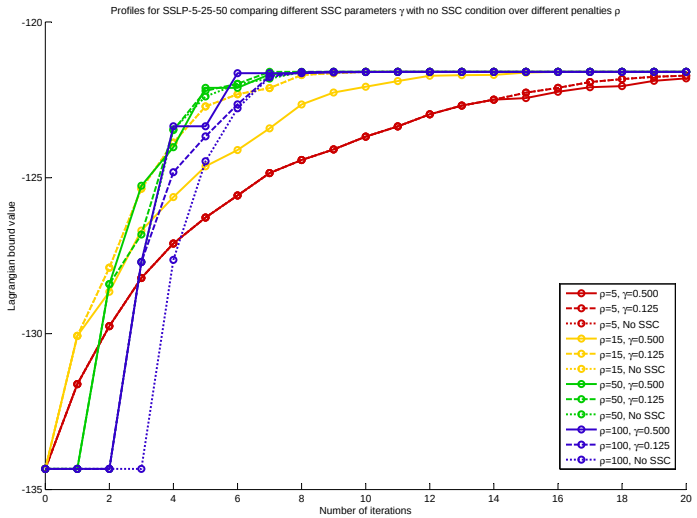

Fig. 3 Applying SDM-GS-ALM to SSLP-5-25-50 using different parameterizations for the SSC condition (or none).

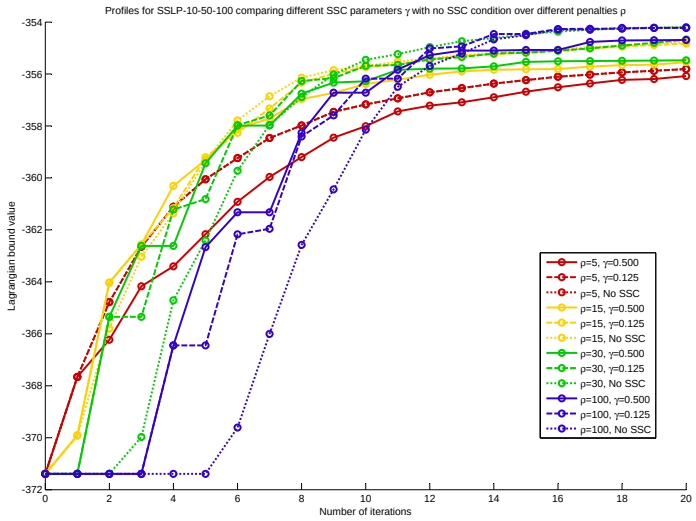

Fig. 4 Applying SDM-GS-ALM to SSLP-10-50-100 using different parameterizations for the SSC condition (or none).

C Supplementary Material: Additional Tables

For each entry (A, B) of Tables 3 and 4, A provides the number of iterations at termination, and B provides the average wall clock time (in seconds) per iteration. The wall clock times for the OOQP and PIPS-IPM experiments are copied from [41]. The experiments in [41] were carried out on the Fusion computing

	SSLP 5-25-100			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(8, 6.31)	(8, 6.33)	(8,3.22)	(8,3.32)
8	(8, 1.14)	(8, 1.21)	(8,0.74)	(8,0.69)
16	(8, 0.71)	(8, 0.74)	(8,0.49)	(8,0.47)
32	(8, 0.54)	(8, 0.53)	(8,0.39)	(8,0.37)
	SSLP 10-50-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(26, 3301)	(22, 2939)	(30, 168.80)	(29, 171.85)
8	(31, 1252)	(24, 1049)	(30, 24.58)	(29, 24.71)
16	(27, 1224)	(28, 1005)	(30, 13.04)	(30, 13.39)
32	(31, 1106)	(27, 865)	(30, 7.79)	(28, 8.19)
	SSLP 10-50-2000			
No. Proc.	SDM-GS1-ALM		SDM-GS5-ALM	
1	(20, 840.69)		(20, 845.77)	
2	(20, 359.63)		(20, 361.40)	
4	(20, 174.83)		(20, 175.03)	
8	(20, 90.51)		(20, 91.46)	
16	(20, 44.98)		(20, 45.76)	
32	(20, 24.27)		(20, 24.09)	
64	(20, 13.87)		(20, 13.88)	

Table 3 SSLP: Reporting auxiliary data

	DCAP 233-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(68, 16.15)	(66, 12.71)	(68, 3.67)	(68, 5.21)
8	(68, 6.62)	(70, 2.39)	(68, 0.53)	(68, 0.64)
16	(68, 5.75)	(73, 1.56)	(68, 0.28)	(68, 0.33)
32	(68, 9.91)	(70, 1.24)	(68, 0.16)	(68, 0.19)
	DCAP 243-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(57, 14.37)	(57, 12.11)	(57, 3.72)	(57, 5.11)
8	(57, 5.04)	(58, 2.12)	(57, 0.57)	(57, 0.67)
16	(57, 4.00)	(59, 2.07)	(57, 0.30)	(57, 0.35)
32	(57, 7.26)	(59, 1.88)	(57, 0.17)	(57, 0.20)
	DCAP 332-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(82, 13.51)	(80, 9.45)	(82, 2.94)	(82, 4.85)
8	(82, 6.65)	(79, 1.70)	(81, 0.43)	(82, 0.57)
16	(82, 5.81)	(80, 1.89)	(81, 0.23)	(82, 0.30)
32	(82, 11.20)	(77, 1.43)	(82, 0.13)	(82, 0.21)
	DCAP 342-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(59, 14.78)	(71, 12.07)	(59, 3.80)	(59, 5.57)
8	(59, 6.03)	(67, 3.19)	(59, 0.53)	(59, 0.68)
16	(59, 5.46)	(56, 2.77)	(59, 0.29)	(59, 0.36)
32	(59, 8.05)	(62, 2.60)	(59, 0.17)	(59, 0.21)

Table 4 DCAP: Reporting auxiliary data